Trabajo de grado en modalidad de aplicación

[173022] Diseño de una metaheurística GRASP para el problema de la programación de la producción en una empresa del sector cerámico bajo un entorno *Distributed Permutation Flow Shop with Flowline Elegibility*, que minimice la tardanza total teniendo en cuenta la importancia del producto y del cliente

Jenny Paola Joya López[a,c] , Paula Alejandra Llorente Castiblanco[a,c]
Oscar David Barrera Ferro[b,c], Eliana Maria Gonzalez Neira[b,c]

[a]*Estudiante de Ingeniería Industrial*
[b]*Profesor, Director del Proyecto de Grado, Departamento de Ingeniería Industrial*
[c]*Pontificia Universidad Javeriana, Bogotá, Colombia*

This research project aims to solve the scheduling problem in an industry of the ceramic sector which objective is the minimization of total tardiness. This problem takes place in a leading floor and wall covering company, where the total tardiness refers to the total delay time of a job delivered after the dates negotiated with customers. At present, the company presents high non-compliance levels with respect to the date of delivery of manufacturing orders, reflected in 935 days of total tardiness during the first quarter of the year. This delay can be improved through different engineering techniques for production scheduling, thus creating a better solution as evidenced in previous scientific studies.The system studied is a Distributed Permutation Flow Shop with Flowline Elegibility (DPFSFE) that considers products and customers importance. To solve the problem a mixed integer linear programming model is proposed in a first stage. Later, due to the NP- hardness of the problem, with a GRASP metaheuristic hybridized with PAES and AHP methodologies was designed and implemented. In order to evaluate the impact of the proposed technique, the results were compared with those obtained with the implementation of the dispatching rule Apparent Tardiness Cost with Setups (ATCS).

The proposed metaheuristic (Figure 1) improves the actual scheduling of the company in a 67% and is a 76% better in comparison with ATCS rule, in terms of total tardiness.

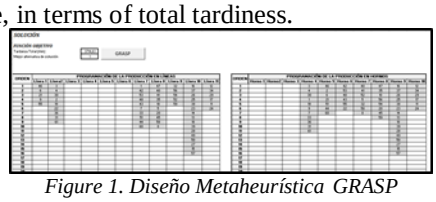


Figure 1. Diseño Metaheurística GRASP

Palabras claves: Programación de la producción, Distributed Permutation Flow Shop, Variables cualitativas, Programación Lineal, Metaheurística GRASP.

1. Justificación y planteamiento del problema

La programación de la producción es una de las actividades, a nivel operativo, más importantes de una empresa que busca ser competitiva en el mercado. Se relaciona con la optimización de varias medidas de rendimiento, tales como: el uso eficiente de los recursos, la entrega de los productos en plazos determinados y la reducción de costos de producción, todo esto encauzado al buen funcionamiento del sistema y a la satisfacción de los clientes (Fuchigami & Rangel, 2017). En este orden de ideas la complejidad de la programación de la producción radica en la variedad de productos de la empresa, la dificultad del proceso, la complejidad de la

estructura de los productos, la susceptibilidad a cambios en la programación (Zandin, 2001), la variabilidad del sistema y la necesidad de alcanzar óptimos niveles de servicio.

Dada la complejidad del tema, se ha evidenciado un progreso importante en el diseño de herramientas de solución para problemas de programación de la producción. Por ejemplo, en un caso real surgido en la empresa BASF AG en Ludwigshafen, Alemania, la programación de trabajos se llevó a cabo a partir de las siguientes metodologías para minimizar el tiempo de finalización ponderado promedio: programación lineal, relajaciones combinatorias, Branch and bound, búsqueda local, modelos heurísticos y relaciones entre ellas. Los resultados experimentales representan un ejemplo de buena sinergia entre la teoría, la experimentación y la práctica al mejorar el escenario real de la empresa (Savelsbergh, Uma, & Wein, 2005). Otro ejemplo del impacto de estas herramientas se demuestra por medio de un estudio realizado en una empresa del sector industrial, que trabaja bajo un esquema de producción Hybrid Flow Shop con recirculación y que tiene como objetivo minimizar el número (ponderado) de trabajos tardíos a través de un Algoritmo Genético (GA). Con el fin de evaluar la efectividad del estudio, se compararon los resultados de la propuesta del GA con un software comercial, lo que mostró que la propuesta del GA tuvo un buen desempeño en la asignación de recursos, mejorando la solución del software de la empresa en un 42%. Sin embargo, cuando se tomó como solución inicial del GA la solución dada por el software de la empresa, se lograron mejoras del 50% (Desprez, Chu, & Chu, 2009).

Teniendo como referencia el impacto positivo generado por las herramientas de solución para problemas de programación, el presente caso de investigación surge a partir de la necesidad de realizar la programación de la producción de trabajos que minimice la tardanza total en una empresa dedicada a la manufactura de baldosas para pisos y paredes. Los trabajos para programar constan de baldosas con una referencia y formato específico de acuerdo con el portafolio de la compañía, el cual ofrece aproximadamente 8 formatos y 268 referencias de baldosas.

El sistema de producción de la empresa consta de cuatro etapas primordiales: la línea de ensamble, la movilización, el horneado y la revisión final. En primer lugar, las 11 líneas de ensamble de la planta cuentan con 4 procesos principales: prensado, secado, esmaltado y decorado, los cuales están comunicados por medio de líneas de transporte. Nueve de las líneas de ensamble cuentan cada una con una única máquina prensadora, mientras que las otras dos líneas cuentan cada una con dos máquinas prensadoras que las alimentan. Cabe resaltar que las líneas no se comunican entre ellas, y que las máquinas dentro de una misma línea están conectadas por bandas transportadoras, por ende, cada línea sigue un entorno Permutation Flow Shop (PFS). Sin embargo, las líneas no son totalmente idénticas entre sí, ya que manejan tecnologías diferentes y los formatos y referencias que pueden trabajar son también distintos. Esta última característica hace que haya restricciones de elegibilidad de máquina, en este caso particular, elegibilidad de línea ya que no todos los trabajos podrán hacerse en todas las líneas.

En segundo lugar, la etapa de movilización en donde se almacena temporalmente y se transporta el producto, consta de dos transfer que cargan el material, en carros, de las líneas a la zona de almacenamiento y lo descargan en los hornos, comunicando así ambos procesos. Sin embargo, no todos los Transfer alimentan todos los hornos ni reciben material de todas las líneas, por ende, los tiempos de desplazamiento entre las diferentes combinaciones línea-horno varían. Lo anterior implica que haya elegibilidad línea-horno es decir que no todas las líneas pueden alimentar todos los hornos.

En tercer lugar, el proceso de cocción en donde se aumenta la resistencia del producto y se otorga el acabado deseado representa para la empresa, según la teoría de restricciones (TOC), el proceso cuello de botella. El proceso de cocción de los diferentes formatos puede llevarse a cabo en cualquier horno, sin embargo, su viabilidad está determinada por el ancho útil del canal de este, buscando siempre un porcentaje de utilización superior al 85%. Lo anterior implica que se presente elegibilidad de horno ya que no todos los formatos de los trabajos alcanzan la mínima eficiencia en los hornos.

Por último, la etapa de revisión final en donde se inspecciona, separa y empaca el producto de acuerdo con criterios de calidad establecidos para su posterior distribución, finaliza el proceso de manufactura de baldosas. Adicionalmente, cada horno está asociado a un único puesto de revisión final lo que conforma el segundo PFS.

Teniendo en cuenta que cada Flow shop está conformado por un gran número de máquinas, el diseño que se trabaja a continuación las agrupa para cada PFS de la siguiente manera: el PFS de las líneas está conformado por

la prensa, el secadero, el equipo de esmaltado y la máquina de cargue al Transfer; mientras el PFS de los hornos se compone de la máquina de descargue al horno, el horno y el puesto de revisión final. Con base en la descripción anterior, cada trabajo debe ser asignado a uno de los PFS de las líneas de ensamble seguido por uno de los PFS de los hornos.

La configuración de las líneas de ensamble corresponde entonces a PFS no idénticos en paralelo, lo cual es similar al problema denominado como Distributed PFS (DPFS) en la literatura. La diferencia principal con el DPFS estudiado por primera vez por (Naderi & Ruiz, 2010), es que los PFS en paralelo aquí estudiados no son idénticos, es decir, los tiempos de procesamiento de los trabajos en cada uno de los PFS en paralelo pueden ser diferentes debido a una tecnología distinta entre máquinas. Además, tienen la característica de que no todos los PFS en paralelo pueden procesar todos los trabajos, lo cual llamaríamos elegibilidad de línea. Por tanto, se denominará la configuración del entorno productivo de la empresa como DPFS with Flowline Elegibility (DPFSFE) tal como ha sido nombrado por (Duan et al., 2016) quienes han sido los primeros en tratar un problema con mayor número de similitudes al aquí expuesto.

De la misma manera que sucede con la etapa de ensamble, los PFS de los hornos se comportan también como DPFSFE. Por tanto, la configuración del entorno productivo en su totalidad corresponde a dos DPFSFE conectados por dos transfer que hacen la distribución entre uno y otro.

Los entornos DPFS fueron estudiados por primera vez por (Naderi & Ruiz, 2010). Los autores proponen los PFS todos idénticos entre sí, de hecho, asumen que cada PFS es una planta de producción diferente. En este trabajo se tiene la diferencia de tener PFS no idénticos al interior de cada DPFS, y que hay dos diferentes DPFS conectados por dos transfer. Por tanto, se deben tomar cuatro tipos de decisiones: i) a cuál de los PFS del DPFS de Ensamble debe asignarse cada trabajo, ii) cuál es la secuencia de procesamiento de los trabajos asignados a cada PFS de las líneas, iii) a cuál de los PFS del DPFS de los hornos debe asignarse cada trabajo y, iv) cuál es la secuencia de procesamiento de los trabajos asignados a cada PFS de la etapa de hornos.

Partiendo del reconocimiento del esquema de la producción actual, se describe la metodología actual que emplea la empresa para la programación de la producción, llevada a cabo en dos fases, la primera táctica y la segunda operacional.

Fase 1. Planeación de los formatos: Mensualmente se reciben los requerimientos de las cantidades en metros a fabricar para los diferentes formatos. Con base en esta solicitud, se decide en qué combinación de líneas y hornos se va a trabajar y cuántos metros por formato se pueden producir en cada una de ellas durante el mes. Esta planeación permite asignar los formatos que se trabajarán en las diferentes plantas de la organización, por tanto, es necesario considerar múltiples escenarios de forma ágil y óptima, garantizando la eficiencia global del proceso.

Esta asignación inicial se realiza manualmente en Excel, con base en la experiencia de la persona encargada, en la capacidad instalada de la planta, en los mantenimientos programados basados en el tiempo (TBM), en los paros programados y en los tiempos de ensayo programados. Todo este proceso tiene como objetivo maximizar el cumplimiento de los indicadores principales de la planta, que miden la eficiencia global del proceso (EGP), disponibilidad, rendimiento y calidad; cumpliendo un valor mínimo establecido para cada uno de ellos.

Fase 2. Programación de trabajos: La planeación resultante de la primera fase es enviada al equipo de programación de la producción que se encarga de determinar el orden de fabricación de los trabajos con el fin de minimizar el tiempo de entrega al cliente.

En primer lugar, se registran los trabajos en un ERP asociándolos a los requerimientos de máquinas de ensamble, este software asigna y agrupa los trabajos al horno que cumple con las especificaciones que requieren los productos. Posteriormente, se analizan los trabajos asignados a cada horno evaluando los inventarios en planta y en el centro de distribución, con ayuda de un segundo software, y se realizan los ajustes necesarios de acuerdo con la cantidad de metros a fabricar para cumplir las órdenes y mantener el stock de seguridad.

Finalmente, con ayuda de un tercer software, se realiza una priorización de trabajos de acuerdo con múltiples criterios que consideran el tipo de cliente (importancia del cliente), el tiempo restante para la entrega del producto y la etapa del ciclo de vida en que se encuentra el mismo (importancia del producto). Luego de este proceso se

debe hacer una validación manual que arrojará la secuencia de producción de las referencias por combinación línea-horno. Esta fase se repite cada dos días dada la fluctuación de la demanda y se considera un tiempo de entrega estándar por trabajo dependiendo la ubicación del cliente.

En la Ilustración 2, se muestran los días de incumplimiento para cada orden de producción con respecto a la fecha de entrega pactada durante el periodo comprendido entre febrero y abril de 2018. Cabe resaltar que los trabajos se organizan cronológicamente lo que permite validar que durante el periodo en estudio se evidencia un constante nivel de incumplimiento en las ordenes de producción, reflejado en una tardanza promedio por trabajo de 2.3 días.

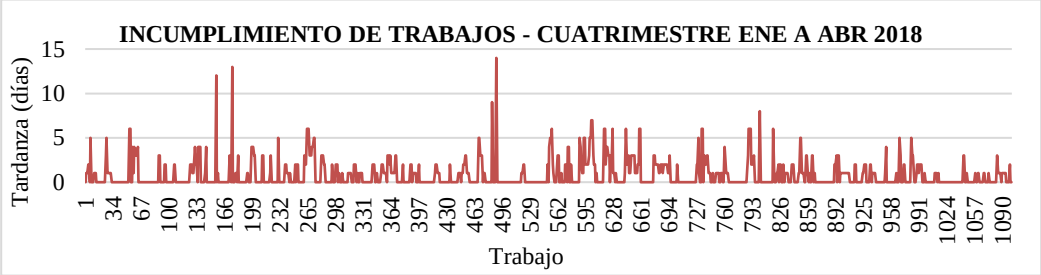


Ilustración 2. Incumplimiento de trabajos del cuatrimestre enero a abril de 2018.

Considerando que se presenta un constante nivel de incumplimiento, que durante el primer cuatrimestre del año 2018 se reporta una tardanza total de 935 días y que existe un alto componente manual durante el proceso de programación de la producción, se constata la necesidad de implementar un método de optimización para la programación de la producción que le permita a la empresa minimizar la tardanza total, considerando las restricciones asociadas.

Diversos métodos exactos, heurísticos y metaheurísticos han sido utilizados para resolver problemas de programación de la producción. Su selección depende del nivel de complejidad del problema a resolver. El problema básico de DPFS propuesto por (Naderi & Ruiz, 2010) para la minimización del makespan tiene complejidad NP-hard (Naderi & Ruiz, 2014), por tanto el uso de métodos exactos tendría sólo sentido en instancias pequeñas dado el alto tiempo computacional que consume. Más bien, se avala el uso de procedimientos heurísticos y metaheurísticos para resolverlos.

En el caso de estudio a continuación se selecciona una metaheurística como técnica de solución. Esta técnica se caracteriza por ser una heurística de propósito general que proporciona una estrategia de alto nivel para obtener soluciones factibles dentro del espacio de búsqueda, haciendo uso de procedimientos eficientes y robustos (McGovern & Gupta, 2011) que tratan de huir de óptimos locales a diferencia de las heurísticas (Sait & Youssef, 2000).

Una de las metaheurísticas que mejor se acomoda al problema actual es GRASP, gracias a su fácil implementación (González & Montoya, 2017) y flexibilidad para adaptarse a diferentes tipos de problemas, como: planificación y programación de la producción, problemas de ubicación, gráficos, asignación cuadrática y lógica (Resende, 1998). Esta metodología es un proceso iterativo en el que cada iteración comprende dos fases: una fase constructiva aleatorizada de naturaleza “greedy” y una fase de búsqueda local. En la fase de construcción, se genera una solución paso a paso. Cada elemento de la solución es evaluado por una función “greedy” e incorporado (o no) en una lista de candidatos restringidos (RCL). El elemento para unir a la solución se elige aleatoriamente de la RCL. Cada vez que se agrega un nuevo elemento a la solución parcial, el algoritmo continúa con la fase de búsqueda local y la solución es actualizada. Todo el proceso se repite hasta que la solución esté completa (Fernandes & Lourenço, 2007).

Para poder abordar la inclusión de los criterios cualitativos de decisión, la importancia del cliente y la importancia del producto se hibridizaron dos metodologías, una multiobjetivo y otra multicriterio. La primera con el fin de identificar las mejores soluciones Pareto evaluándolas en términos de tres objetivos, la tardanza total y las tardanzas totales ponderadas según la importancia del cliente y del producto, y la segunda con el objetivo de seleccionar una única solución de la frontera dependiendo de la importancia que el tomador de

decisiones dé a cada uno de los tres objetivos mencionados. Cabe resaltar que aunque para la compañía los criterios cualitativos son importantes, a la empresa le interesa considerar la tardanza total ya que esta tiene un efecto conveniente en los costos frente a la tardanza total ponderada; adicionalmente, para la organización es imperativo no descuidar las entregas a los clientes con menor importancia al momento de considerar la inclusión de los criterios cualitativos.

Dentro de las metodologías multiobjetivo se encuentran Multi-objective genetic algorithm (MOGA), Strength Pareto evolutionary algorithm (SPEA), Pareto archived evolution strategy (PAES), Nondominated Neighbor Immune Algorithm (NNIA), entre otras. De estas se ha escogido la metodología PAES por su facilidad de implementación, por estar entre los cinco algoritmos evolutivos multiobjetivo con mayor número de artículos, por ser de segunda generación lo cual quiere decir que se centra en la eficiencia computacional y por su concepto de jerarquización (Pinilla & Castro, 2015). Adicionalmente, esta metodología proporciona un campo de acción más amplio y flexible para la empresa con el fin de implementar estrategias comerciales que se acoplen a las necesidades de la compañía en un momento dado. Cabe resaltar que, aunque la tardanza total y la tardanza total ponderada no son totalmente independientes entre sí, tampoco son directamente proporcionales, lo que implica que dos secuencias de programación con tardanza total idéntica no necesariamente tienen la misma tardanza ponderada, por lo tanto entre ellas son comparables.

Por otro lado, dentro de las técnicas que comparan múltiples criterios se encuentran The integral analysis method (IAM) (García, 2007), The surrogate worth trade off method (Haimes, 1998), Analytic Hierarchy Process (AHP), Elimination Et Choix Traduisant la Realité (ELECTRE) (Goodman & Hastak, 2015) y available-to-promise (ATP) model (Jung, 2010). De estas se ha seleccionado AHP porque es un procedimiento simple (Chang & Lo, 2001), potente y completo que permite incorporar factores cualitativos y cuantitativos en el proceso de toma de decisiones, basándose en tres principios fundamentales: la construcción de jerarquías, el establecimiento de prioridades y la coherencia lógica para asegurar que los elementos se agrupen de forma racional (Zandin, 2001).

Finalmente, para medir la calidad del diseño es posible contrastarlo con el modelo óptimo o con metodologías que garanticen, desde la literatura, un buen desempeño en contextos y objetivos similares. Dada la complejidad del problema no es posible comparar el modelo óptimo con la metaheurística a la misma escala. Sin embargo, existen herramientas como metaheurísticas, heurísticas, reglas de despacho, entre otras, que permiten llegar a una buena solución permitiendo ser un punto de comparación. Buscando una herramienta de fácil implementación se selecciona una regla de despacho, en este caso la ATCS, que presenta una mejoría del 30% en la minimización de la tardanza con respecto a la regla despacho Apparent Tardiness Cost (ATC) (Lee, Bhaskaran, & Pinedo, 1997), considera tiempos de alistamiento y presenta resultados interesantes en instancias grandes (Lamothe, Marmier, Dupuy, Gaborit, & Dupont, 2012).

En ese orden de ideas, este trabajo responde la siguiente pregunta de investigación: *¿Cómo diseñar la metaheurística GRASP hibridizada con la metodología PAES y AHP para dar solución al problema de la programación de la producción en la unidad de negocio de pisos y paredes de una empresa del sector cerámico, bajo un entorno de dos DPFSFE conectados, que minimice la tardanza total sin olvidar la importancia del producto y del cliente?*

2. Antecedentes

La revisión de literatura se encuentra dividida en cinco temas que sobresalen en el presente caso de estudio, en primer lugar, lo relativo a problemas Distributed shop, en segundo lugar, lo relacionado con la metaheurística GRASP para solución de problemas de programación de la producción, en tercer lugar, lo referente a la integración de criterios cualitativos, en cuarto lugar, lo vinculado a modelos multiobjetivo y multicriterio en programación y, por último, lo concerniente a reglas de despacho para la programación.

2.1. Problemas Distributed Shop

En el entorno de la producción, se ha constituido un paradigma denominado problema Distributed, en donde la idea tradicional de una sola instalación para llevar a cabo la producción ha ido cambiando a múltiples instalaciones descentralizadas. Las ventajas de los esquemas Distributed son los bajos costos de producción,

los productos de mayor calidad, la flexibilidad de producción y la cercanía a proveedores y clientes, por lo que se tiene mayor velocidad de respuesta a los cambios del mercado (Azab & Naderi, 2014a).

El esquema Distributed ha sido estudiado en entornos Job shop (JS) y Flow Shop (FS), a continuación, se mencionan algunos estudios relacionados con cada uno de ellos. En cuanto al entorno Distributed Job shop (DJS) (Azab & Naderi, 2014b) formularon dos modelos de programación lineal entera mixta y seis heurísticas, de las cuales tres son de naturaleza "Greedy", para dar una solución efectiva al problema DJS. Por su parte, (Lu, Tan, Peng, Chen, & Wu, 2014) propusieron una nueva solución de representación (denominada Snew) y desarrollaron dos nuevos algoritmos metaheurísticos (llamados GA-Snew y ACO-Snew). Los resultados obtenidos a partir de los dos algoritmos propuestos superan algoritmos metaheurísticos anteriores para resolver los problemas de programación de DJS. (Chang & Liu, 2015) propusieron un algoritmo genético híbrido (HGA) para resolver el problema de programación distribuida y flexible. Los resultados experimentales indicaron que el enfoque propuesto es considerablemente robusto, superando los algoritmos previos después de 50 corridas. Por último, cabe resaltar a (Chaouch, Driss, & Ghedira, 2017), quienes dieron una visión general de los estudios pioneros DJS realizados en la solución del Problema de Programación que utilizaron diferentes técnicas y alcanzaron una función objetivo especificada.

Por otro lado, en cuanto al entorno Distributed Flow Shop, se identifica el esquema de Permutation. Este hace referencia a las diferentes combinaciones de secuencias de trabajos que pueden ejecutarse en una máquina, o series de ellas. La programación de la producción en un entorno Distributed Permutation Flow Shop (DPFS) es un problema trabajado recientemente con robustos antecedentes de ingeniería (Gao & Chen, 2011b). El DPFS puede ser visto como una versión generalizada del PFS (Naderi & Ruiz, 2010), entorno estudiado con mayor intensidad en el área de la programación. En este enfoque se ha hecho énfasis en la minimización del makespan, con el objetivo de disminuir el tiempo de ejecución de la producción (Framinan & Leisten, 2003).

(Naderi & Ruiz, 2010) presentaron por primera vez el DPFS. Ellos desarrollaron seis modelos diferentes de programación lineal entera mixta (MILP) para el problema considerado y propusieron dos reglas simples de asignación de FS (fábricas) junto con 14 heurísticas basadas en reglas de despacho, heurísticas constructivas efectivas y métodos Variable Neighborhood Descent (VND). Paralelo a ellos (Liu & Gao, 2010) propusieron un algoritmo para el mismo problema con base en mecanismos electromagnéticos. Un año después, (Gao & Chen, 2011) propusieron un algoritmo basado en el algoritmo genético para minimizar el tiempo máximo de finalización, ajustando el método a un escenario DPFS, denotado Genetic Algorithm with a Local Search (GA-LS) y (Gao & Chen, 2011b) publicaron una heurística constructiva mejorada a través de una nueva regla de despacho, la cual inserta un grupo de trabajos en las fábricas al mismo tiempo, en vez de insertar uno a la vez como en las reglas originales. Adicionalmente, en la literatura existe una investigación del entorno DPFS que considera la elegibilidad de línea, siendo esta la que más se asemeja al caso de estudio en desarrollo, en donde se estudia el entorno DPFSFE. (Duan et al., 2016) estudiaron la extensión mencionada anteriormente buscando minimizar el makespan de todas las fábricas, mediante las heurísticas de (Framinan & Leisten, 2003) y el método de búsqueda del vecino más cercano, obteniendo que ambos métodos son efectivos para resolver el caso en pequeñas y grandes escalas.

2.2. Programación a través de GRASP

Considerando la aplicación de diferentes heurísticas y metaheurísticas para la solución del problema de la programación de la producción se identifica que la metaheurística GRASP es uno de los algoritmos más simples de implementar del cual se han obtenido buenos resultados (González & Montoya, 2017). (Rojanasoonthon & Bard, 2005) demostraron que la metaheurística GRASP produjo mejores soluciones en 20 de 25 instancias de la programación de la producción, bajo un entorno de máquinas en paralelo, en comparación con la heurística Reddy and Brown. Por su parte, (Bierwirth & Kuhpfahl, 2017) construyeron, con base en esta metaheurística y modelos gráficos, una nueva heurística para la programación de la producción en un esquema Job Shop con ventanas de tiempo para minimizar la tardanza total, obteniendo un modelo con los mejores resultados conocidos hasta el momento; (González & Montoya, 2017) aplicaron la metaheurística GRASP en un esquema Hybrid Flow Shop para datos experimentales basados en la literatura, logrando una mejor solución en un 27% con respecto a las reglas de despacho establecidas. Los casos anteriormente expuestos comprueban el buen desempeño de esta metaheurística en problemas de

programación de la producción, sin embargo, no existen en la literatura muchas aplicaciones de este modelo en el entorno PFS y ninguna en el DPFS.

2.3. Criterios cualitativos en programación

Los criterios cualitativos en temas de programación han sido poco estudiados, sin embargo como (González & Montoya, 2018) manifiestan, son de gran importancia al permitir reducir la brecha entre la teoría y la práctica. En esta ponencia, los autores proponen un enfoque de optimización multicriterio para resolver un Stochastic Permutation Flow Shop (SPFS) que incluye criterios de decisión tanto cuantitativos como cualitativos, con el fin de obtener una programación más robusta.(González, García, Caballero, Molina, & Montoya, 2016) se ocuparon de la programación en un problema de doble criterio bajo un esquema Stochastic Flexible Flow Shop (SFFS) donde existe un criterio cuantitativo (la tardanza total ponderada) y otro cualitativo (la importancia del cliente para la empresa). Para resolver este problema, aplicaron el método IAM. Los resultados mostraron que IAM permite la selección de las alternativas que logran de la mejor manera ambos tipos de criterios.

Años previos, (Jung, 2010) presentó el modelo ATP considerando dos factores: la prioridad del cliente y la variación de los costos de penalización. Por su parte, (Chang & Lo, 2001) propusieron una función multicriterio que incluía tres aspectos cualitativos: las consideraciones de marketing, la importancia estratégica de los clientes y el beneficio / riesgo de la orden en un entorno Job Shop. Con el fin de dar solución a este problema se utilizó un hibrido entre el algoritmo genético, la búsqueda tabú y el AHP; así como el algoritmo de la colonia de hormigas desarrollado por (Chang, Lin, Pai, Zhong, & Hung, 2008). Los resultados de ambas técnicas de solución presentan resultados potentes y comparables, sin embargo, este último proporciona un menor tiempo computacional.

2.4. Métodos multiobjetivo y multicriterio en programación

A fin de considerar modelos multiobjetivo, (Knowles & Corne, 1999) introdujeron un esquema de evolución simple llamado PAES, en el cual muestran la obtención de un algoritmo no trivial que permite construir la frontera de Pareto. Años después, (Armentano & Arroyo, 2004) proponen un algoritmo de búsqueda tabú para dar solución a problemas multiobjetivo generando varias soluciones en el conjunto de óptimos de Pareto. El método propuesto fue comparado con el algoritmo Branch-and-Bound propuesto por (Daniels & Chambers, 1990) y el algoritmo genético de (Ishibuchi & Murata, 1998), obteniendo mejores resultados. Contemporáneamente, (Varadharajan & Rajendran, 2005) desarrollaron una heurística para solucionar el problema de la programación, bajo un entorno FS, con el objetivo de minimizar el makespan y el tiempo de flujo de trabajo, mediante la técnica de simulated annealing.

Por su parte, (Arroyo & De Souza Pereira, 2011) desarrollaron una metaheurística GRASP para dar solución al problema en mención, bajo un esquema FS, orientado a optimizar de dos a tres objetivos simultáneamente; mientras (Lin, Lee, & Wu, 2012) proponen una combinación entre el AHP y el algoritmo genético (GA) para resolver el problema de programación de producción dinámica bajo un esquema FS con múltiples criterios. El GA se aplica con el objetivo de obtener programaciones cercanas a las óptimas mientras la metodología AHP funciona como herramienta doble efecto, para solucionar el problema multicriterio y como elemento de velocidad de convergencia para el GA. Los autores mostraron resultados consistentes y con mejores resultados que las metodologías manuales.

Años más tarde, (Martí, Campos, Resende, & Duarte, 2015) proponen hibridizar la metaheurística GRASP con el Path Relinking para solucionar problemas de optimización multiobjetivo. Técnicas comprobadas a través de 70 instancias y 30 algoritmos, que muestran que las heurísticas propuestas son competitivas con los métodos planteados para estos problemas. Finalmente, como metodología para resolver el problema SPFS considerando criterios cualitativos (González & Montoya, 2018) proponen una semiheurística basada, principalmente, en la Metaheurística GRASP en la cual integran la simulación de Monte Carlo para trabajar la estocasticidad del modelo, el algoritmo PAES para tratar los objetivos múltiples y la metodología AHP para calificar todas las soluciones Pareto.

2.5. Reglas de despacho para la programación de la producción

A través de los años, muchas reglas de despacho han sido propuestas por varios investigadores. (Haupt, 1989) las clasificó en reglas basadas en el tiempo de procesamiento, en la fecha de entrega, la combinación entre ambas y otras. Un clásico ejemplo del primer grupo es la regla Shortest Process-Time (SPT) que busca minimizar el tiempo de flujo promedio. (Low & Lin, 2011) encontraron la secuencia óptima con base en esta regla para el problema de la programación en una sola máquina con el efecto de aprendizaje dependiente del tiempo. Una muestra de la segunda clasificación es la regla Earliest Due Date (EDD) en donde se da prioridad a los trabajos con tiempos de entrega más próximos a vencer. (Roychowdhury, Allen, & Allen, 2017) aplicaron esta regla para el problema de la programación en una empresa de estampado donde hallaron optimalidad en instancias que consideraban inventarios iniciales altos y sin tardanza, y soluciones regulares en instancias con inventarios iniciales bajos y horizontes de planeación largos. Critical Ratio (CR) es una de las reglas que combina el tiempo de procesamiento y la fecha de entrega para indicar el grado de urgencia del trabajo (Chiang & Fu, 2004). En el estudio de (Shafaei & Brunn, 1999), CR fue la segunda mejor regla para la minimización de costos considerando la tardanza y los costos de mantener inventario. Adicionalmente, en esta categoría resaltan reglas como Cost Over Time (COVERT) y ATC que se desempeñan muy bien en la minimización de trabajos tardíos esperados en presencia de bajos de coeficientes de variación (González, Montoya, & Caballero, 2018) y se destacan por su competitividad bajo diversos objetivos de desempeño en FS y JS. La desventaja de las reglas COVERT y ATC es que son reglas parametrizadas, y su rendimiento depende de los valores elegidos para el parámetro k (El Bouri & Amin, 2015). Cabe añadir que una extensión del ATC es el ATCS propuesto por (Lee et al., 1997), el cual incluye tiempos de alistamiento. Por otro lado, en la última categoría se pueden clasificar reglas como PT+PW+ODD y PT+WINQ+SLACK. En sus estudios (Rajendran & Holthaus, 1999) y (Jayamohan & Rajendran, 2000) obtuvieron buen desempeño en la primera regla minimizando la tardanza promedio y la tasa de tardíos y buen desempeño en la segunda minimizando la tardanza máxima.

3. Objetivos

Diseñar la metaheurística GRASP hibridizada con la metodología PAES y AHP para dar solución al problema de la programación de la producción en la unidad de negocio de pisos y paredes de una empresa del sector cerámico, bajo un entorno Distributed Permutation Flow Shop with Flowline Elegibility, que minimice la tardanza total, sin olvidar la importancia del producto y del cliente.

- Formular un modelo matemático de programación lineal que represente el problema de la programación de la producción en la empresa, sin tener en cuenta los criterios cualitativos.
- Diseñar e implementar en un lenguaje de programación la metaheurística GRASP hibridizada con la metodología PAES y AHP para resolver el problema de la programación de la producción en la empresa.
- Medir el impacto de la herramienta diseñada en términos de la tardanza total.

4. Metodología

La metodología utilizada en el desarrollo del trabajo se basa en la herramienta DMAIC de la filosofía seis-sigma, la cual consta de cinco etapas: Definición, Medición, Análisis, Mejoramiento y Control (Garza, González, Rodríguez, & Hernández, 2016). En primer lugar, en la etapa de Definición se identifica el problema del incumplimiento de las fechas de entrega pactadas para cada trabajo, posteriormente en la etapa de medición se valida dicha problemática, con lo cual se evidencia que durante el primer cuatrimestre del año 2018 la empresa presenta en promedio 2.3 días de tardanza por trabajo y 935 días de tardanza total. En tercer lugar, en la etapa de análisis se construye el estado del arte, se reconoce el sistema productivo a detalle y se identifican las restricciones que se deben considerar en la herramienta propuesta. Posteriormente, en la etapa de Mejoramiento se proponen tres fases (Ilustración 3): la primera, relacionada con la formulación de un modelo matemático (Objetivo específico 1), la segunda, asociada con el diseño e implementación de la metaheurística GRASP hibridizada con la metodología PAES y AHP para la minimización de la tardanza total (Objetivo específico 2) y la tercera, orientada a medir el impacto de la herramienta diseñada con respecto a la regla de despacho ATCS (Objetivo específico 3). Finalmente, para la etapa de Control se propone medir y validar el impacto de la técnica diseñada con respecto al desempeño real que tuvo la empresa en el primer cuatrimestre del año 2018.

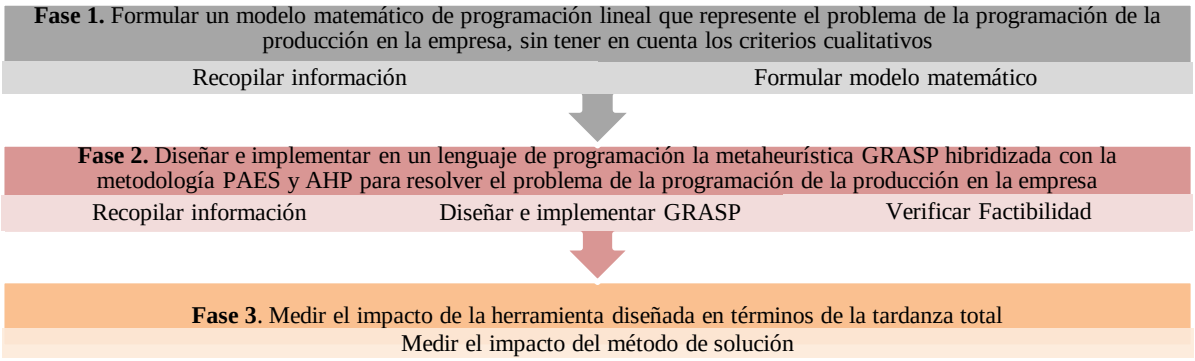


Ilustración 3. Metodología

4.1. Fase 1 – Modelo Matemático

El modelo matemático representa el esquema de producción actual de la empresa en cuestión, ilustración 4. Se tienen |J| trabajos que deben ser programados en una combinación línea – horno factible, es decir, con la capacidad de trabajar en conjunto. Cada línea y horno siguen un esquema de producción PFS con m máquinas, por las cuales debe ser procesado cada trabajo j. Cabe resaltar que todos los PFS de líneas $l \in$ L tienen la misma cantidad de máquinas ml, al igual que los PFS de hornos $h \in$ H que tienen la misma cantidad de máquinas mh.

Cada orden de fabricación o trabajo j está compuesto por un número de baldosas determinado, lo cual implica que los tiempos de finalización en este esquema de producción se comporten de manera diferente a lo presentado comúnmente en la literatura. Con base en este esquema, un trabajo puede ser procesado al mismo tiempo en las m máquinas de un FS, sin embargo, una baldosa puede ocupar en el mismo instante de tiempo una sola máquina. Una vez la primera baldosa del trabajo sale de la primera máquina m, puede comenzar a ser procesada en la máquina siguiente $m + 1$, a partir de este instante las baldosas subsiguientes salen de la primera máquina m a una tasa definida.

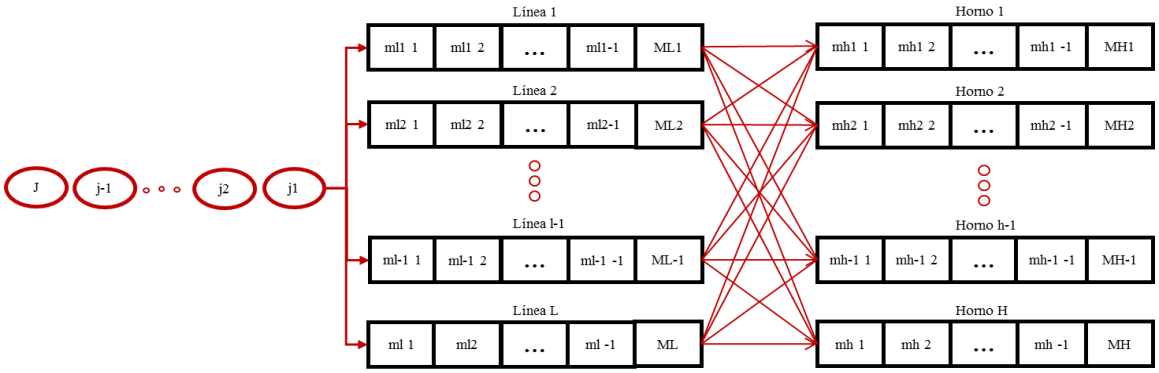


Ilustración 4. Esquema de producción actual de la empresa en cuestión

A continuación, se presenta en detalle la formulación del modelo de programación lineal entera mixta. Se plantean ocho conjuntos (Tabla 1), veintitrés parámetros (Tabla 2), quince variables de decisión (Tabla 3) y 38 restricciones (Ecuaciones 2 a 39).

Símbolo	Conjunto	Símbolo	Conjunto
J	Trabajos que programar	MH	Cantidad de máquinas en hornos
L	Líneas	K	Posiciones
ML	Cantidad de máquinas en las líneas de producción	C	Clientes
H	Hornos	P	Prioridad de producto

Tabla 1 Conjuntos Modelo Matemático

Símbolo	Parámetro
$conex_{lh}$	Binario, 1 si la línea $l \in L$ puede trabajar con el horno $h \in H$, 0 de lo contrario
d_j	Tiempo (min) acordado con el cliente para la entrega del trabajo $j \in J$
r_j	Tiempo (min) en que el trabajo $j \in J$ está disponible para ser programado
$cliente_{cj}$	Binario 1, si el trabajo $j \in J$ tiene asignado el cliente $c \in C$, 0 de lo contrario
$producto_{pj}$	Binario 1, si el trabajo $j \in J$ tiene asignado el producto con prioridad P, 0 de lo contrario
M	Número muy grande
$w\_cliente_c$	Importancia del cliente $c \in C$ para la empresa
$w\_producto_p$	Importancia del producto $p \in P$ para la empresa
on_l_l	Binario, 1 si para el horizonte de programación la línea $l \in L$ está en funcionamiento, 0 de lo contrario
$elegi\_l_{jl}$	Binario, 1 si el trabajo $j \in J$ se puede procesar en la líneal$\in L$, 0 de lo contrario
$procesam\_l_{jl}$	Tiempo (min) de procesamiento del trabajo $j \in J$ en la línea $l \in L$
$arr\_ml\_l_{j\,ml\,l}$	Tiempo (min) de arranque del trabajo $j \in J$ en la máquina $ml \in ML$ de la línea $l \in L$
to_l_l	Tiempo (min) en el que la línea $l \in L$ está disponible para procesar trabajos al comenzar la programación
cp_carros_{jl}	Capacidad de un carro para almacenar baldosas del formato del trabajo $j \in J$ en la línea $l \in L$
$posibles\_cambios\_l_{ij}$	Binario, 1 si hay cambio de formato y/o referencia entre los trabajos $i \in J$ y $j \in J$, en caso de ser programados consecutivamente en líneas, 0 de lo contrario
$temp\_camb\_l_{ij}$	Tiempo (min) para cambio de formato y/o referencia entre los trabajos $i \in J$ y $j \in J$, en caso de ser programados consecutivamente en líneas
on_h_h	Binario, 1 si para el horizonte de programación el horno $h \in H$ está en funcionamiento, 0 de lo contrario
$elegi\_h_{jh}$	Binario, 1 si el trabajo $j \in J$ se puede procesar en el horno $h \in H$, 0 de lo contrario
$procesam\_h_{jh}$	Tiempo (min) de procesamiento del trabajo $j \in J$ en el horno $h \in H$
$arr\_mh\_h_{j\,mh\,h}$	Tiempo (min) de arranque del trabajo $j \in J$ en la máquina $mh \in MH$ del horno $h \in H$
to_h_h	Tiempo (min) en el que el horno $h \in H$ está disponible para procesar trabajos al comenzar la programación
$posibles\_cambios\_h_{ij}$	Binario, 1 si hay cambio de formato entre los trabajos $i \in J$ y $j \in J$, en caso de ser programados consecutivamente en hornos, 0 de lo contrario
$temp\_camb\_h_{ij}$	Tiempo (min) para cambio de formato entre los trabajos $i \in J$ y $j \in J$, en caso de ser programados consecutivamente en hornos

Tabla 2 Parámetros Modelo Matemático

Símbolo	Variable
t_j	Tardanza del trabajo $j \in J$
lh_asig_{jlh}	Binario, 1 si el trabajo $j \in J$ se procesa por la línea $l \in L$ y el horno $h \in H$, 0 de lo contrario
x_{jkl}	Binario, 1 si el trabajo j$\in J$ se procesa en la posición $k \in K$ de la línea $l \in L$
$fin\_l_{j\,ml\,l}$	Tiempo de finalización (min) del trabajo $j \in J$, en la máquina $ml \in ML$ de la línea $l \in L$
$cambio\_real\_l_{ijl}$	Binario, 1 si hay cambio de formato y/o referencia entre los trabajos $i \in J$ y $j \in J$ programados consecutivamente en líneas, 0 de lo contrario
$ini\_k1\_m1_{jl}$	Tiempo de inicio (min) del trabajo $j \in J$ programado en la primera posición de la línea $l \in L$
$ini\_k2\_m1\_0c_{jl}$	Tiempo de inicio (min) del trabajo $j \in J$ programado desde la segunda posición de la línea $l \in L$, cuando el trabajo programado en la posición inmediatamente anterior no requiere cambio de referencia y/o formato
$ini\_k2\_m1_{jl}$	Tiempo de inicio (min) del trabajo $j \in J$ programado desde la segunda posición de la línea $l \in L$, cuando el trabajo programado en la posición inmediatamente anterior requiere cambio de referencia y/o formato
y_{jkh}	Binario, 1 si el trabajo $j \in J$ se procesa en la posición $k \in K$ del horno $h \in H$
$fin\_h_{j\,mh\,h}$	Tiempo de finalización (min) del trabajo $j \in J$, en la máquina $mh \in MH$ del horno $h \in H$
$cambio\_real\_h_{ijh}$	Binario, 1 si hay cambio de formato entre los trabajos $i \in J$ y $j \in J$ programados consecutivamente en hornos, 0 de lo contrario
$ini\_k1\_mh_{jh}$	Tiempo de inicio (min) del trabajo $j \in J$ programado en la primera posición del horno $h \in H$
$ini\_k2\_mh_{jh}$	Tiempo de inicio (min) del trabajo $j \in J$ programado desde la segunda posición del horno $h \in H$
$procesam\_total_{j\,mh\,h}$	Tiempo de procesamiento (min) del trabajo $j \in J$ programado desde la segunda posición del horno $h \in H$

Tabla 3 Variables Modelo Matemático

La función objetivo del modelo busca minimizar la tardanza total obtenida de la programación, ecuación (1).

$$\min z = \sum_{j\ in\ J} t_j \qquad (1)$$

Las restricciones del modelo se dividen de la siguiente manera: conjunto de restricciones generales (ecuaciones 2 a 6), conjunto de restricciones en líneas (ecuaciones 7 a 22) y conjunto de restricciones en hornos (ecuaciones 23 a 39).

$$lh\_asig_{jlh} \leq conex_{lh} \qquad \forall\ j\ \in\ J, l \in L, h \in H \qquad (2)$$

$$\sum_{k\ in\ K} y_{jkh} \leq \sum_{l\ in\ L} lh\_asig_{jlh} \qquad \forall\ j \in J, h \in H \qquad (3)$$

$$\sum_{k\ in\ K} x_{jkl} \leq \sum_{h\ in\ H} lh\_asig_{jlh} \qquad \forall\ j \in J, l \in L \qquad (4)$$

$$\sum_{h\ in\ H, l\ in\ L} lh\_asig_{jlh} = 1 \qquad \forall\ j \in J \qquad (5)$$

$$t_j \geq fin\_h_{j|MH|h} - d_j - M\left(1 - \sum_{k\ in\ K} y_{jkh}\right) \qquad \forall\ j \in J, h \in H \qquad (6)$$

En vista de que los PFS de líneas y hornos son independientes, ya que no todos son aptos para trabajar todas las ordenes de fabricación y que no todas las líneas pueden trabajar con todos los hornos, se plantea el conjunto de restricciones (2) con el cual si una línea l y horno h pueden trabajar juntos, el trabajo a programar puede ser asignado a dicha combinación, de lo contrario no. Con el mismo fin, los conjuntos de restricciones (3) y (4) consiguen que, si un trabajo es asignado a un horno o línea respectivamente, este debe ser asignado a una posición k de los FS correspondientes. Por otro lado, el conjunto (5) garantiza que todo trabajo sea asignado a una combinación línea-horno con capacidad de trabajar juntos. Finalmente, el conjunto de restricciones (6) calcula el tiempo en que fue entregado tarde el trabajo j, en caso de estarlo.

$$\sum_{j\ in\ J} on\_l_l\ x_{jkl} \leq 1 \qquad \forall\ k \in K, l \in L \qquad (7)$$

$$\sum_{k\ in\ K} on\_l_l\ x_{jkl} \leq 1 \qquad \forall\ j \in J, l \in L \qquad (8)$$

$$\sum_{k\ in\ K, l\ in\ L} on\_l_l\ x_{jkl} = 1 \qquad \forall\ j \in J \qquad (9)$$

$$on\_l_l\ x_{jkl} \leq elegi\_l_{jl}\ on\_l_l \qquad \forall\ j \in J, k \in K, l \in L \qquad (10)$$

$$\sum_{j\ in\ J} on\_l_l\ x_{jkl} \geq \sum_{j\ in\ J} on\_l_l\ x_{j(k+1)l} \qquad \forall\ k \in K, l \in L, k < |K| \qquad (11)$$

$$ini\_k1\_m1_{jl}\ \geq \sum_{k\ in\ K} r_j\ x_{jkl} \qquad \forall\ j \in J, l \in L \qquad (12)$$

$$ini\_k1\_m1_{jl} \geq \sum_{k\ in\ K} to\_l_l\ x_{jkl} \qquad \forall\ j \in J, l \in L \qquad (13)$$

$$fin\_l_{j1l}\ \geq\ ini\_k1\_m1_{jl} + arr\_ml\_l_{j1l}\ x_{j1l} + procesam\_l_{jl}\ x_{j1l} \qquad \forall\ j \in J, l \in L \qquad (14)$$

$$fin\_l_{j\ ml\ l} \geq\ fin\_l_{j(ml-1)l} + \sum_{k\ in\ K} arr\_ml\_l_{j1l}\ x_{jkl} \qquad \forall\ j \in J, l \in L, ml \in ML, ml > 1 \qquad (15)$$

$$X_{jkl} + x_{i(k+1)l} \leq cambio\_real\_l_{jil} + 1 \qquad \forall\ k \in K, j \in J, i \in J, l \in L, k < |K|, i <> j, posibles\_cambios\_l_{ji} = 1 \qquad (16)$$

$$ini\_k2\_m1\_0c_{jl} + M\left(1 - x_{jkl}\right) \geq rj_j \qquad \forall\ l \in L, j \in J, k \in K, k > 1 \qquad (17)$$

$$ini\_k2\_m1\_0c_{jl} + M\left(1 - x_{jkl}\right) \geq fin\_l_{i1l} - arr\_ml\_l_{i1l} - M\left(1 - x_{i(k-1)l}\right) \qquad \forall\ i \in J, l \in L, j \in J, k \in K, i <> j, k > 1 \qquad (18)$$

$$fin\_l_{j1l} + M\left(1 - x_{jkl}\right) \geq ini\_k2\_m1\_0c_{jl} + procesam\_l_{jl} - M\left(1 - x_{i(k-1)l}\right) - M\left(cambio\_real\_l_{ijl}\right) \qquad \forall\ i \in J, l \in L, j \in J, k \in K, i <> j, k > 1 \qquad (19)$$

$$ini\_k2\_m1_{jl} + M\left(1 - x_{jkl}\right) \geq r_j - M\left(1 - x_{i(k-1)l}\right) \qquad \forall\ i \in J, l \in L, j \in J, k \in K, i <> j, k > 1 \qquad (20)$$

$$ini\_k2\_m1_{jl} + M\left(1 - x_{jkl}\right) \geq fin\_l_{i4l} + temp\_camb\_l_{ij} - M(1 - x_{i(k-1)l}) \qquad \forall\ i \in J, l \in L, j \in J, k \in K, i <> j\ , k > 1 \qquad (21)$$

$$fin\_l_{j1l} + M\left(1 - x_{jkl}\right) \geq ini\_k2\_m1_{jl} + procesam\_l_{jl} + arr\_ml\_l_{j1l} - M(1 - x_{i(k-1)l}) - M(1 - cambio\_real\_l_{ijl}) \quad \forall\, i \in J, l \in L, j \in J, k \in K, i <> j, k > 1 \qquad (22)$$

Con respecto al conjunto de restricciones del FS de líneas (conjunto 7 a 22), las cinco primeras garantizan una asignación factible mientras las restantes consideran los diferentes casos para el cálculo del tiempo de finalización de los trabajos a programar. El conjunto de restricciones (7) exige que a cada posición k de la línea l solo se le asigne un único trabajo j, mientras el conjunto (8) evita que un trabajo j sea asignado a una línea l en más de una posición k. Por otro lado, el conjunto de restricciones (9) garantiza que cada trabajo j sea asignado exactamente una vez a una posición k y a una línea l.

Con el objetivo de evitar que un trabajo j sea asignado a una línea l que no tenga los requerimientos para procesarlo, se dispone del conjunto de restricciones (10). Adicionalmente, en aras de que las ordenes de fabricación siempre ocupen posiciones continuas a partir de la primera posición, se plantea el conjunto (11), con el cual si la posición k de la línea l no tiene trabajo j asignado la siguiente posición $k + 1$ tampoco lo tendrá.

Con relación a las restricciones asociadas al tiempo de finalización de los trabajos asignados a las líneas de producción, la Tabla 4 detalla los casos para tener en cuenta.

Caso		Descripción
1	K | J | $temp\_camb\_l_{jj}$ | $ML=1$ | $ML=2$ | $ML=3$ | $ML=4$ 1 | 2 | - 2 | 3 | 240 3 | 1 | 10 4 | 4 | 0	Los conjuntos de restricciones (12), (13) y (14) hacen alusión a este caso, en el cual el tiempo de finalización en la máquina 1 del trabajo *j* programado en la posición $k = 1$ depende del máximo entre el tiempo en que está disponible la línea l y el trabajo *j*, más el tiempo de procesamiento y arranque de dicho trabajo en la máquina 1
2	K | J | $temp\_camb\_l_{jj}$ | $ML=1$ | $ML=2$ | $ML=3$ | $ML=4$ 1 | 2 | - 2 | 3 | 240 3 | 1 | 10 4 | 4 | 0	El conjunto de restricciones (15) hace alusión al caso en el cual se calcula el tiempo de finalización del trabajo *j* programado en las posiciones $k \geq 1$ en todas las máquinas a excepción de la primera ($ml <> 1$). Esta variable depende del tiempo de finalización del mismo trabajo en la máquina anterior ($ml-1$) más el tiempo de arranque de dicho trabajo *j* en la máquina en estudio (ml)
3	K | J | $temp\_camb\_l_{jj}$ | $ML=1$ | $ML=2$ | $ML=3$ | $ML=4$ 1 | 2 | - 2 | 3 | 240 3 | 1 | 10 4 | 4 | 0	Los conjuntos de restricciones (17), (18) y (19) hacen alusión a este caso, en el cual el tiempo de finalización en la máquina 1 del trabajo *j* programado en las posiciones $k > 1$ depende del máximo entre el tiempo en que está disponible el trabajo *j* y la suma del tiempo de finalización en la última máquina del trabajo asignado a la posición anterior $k - 1$ con el tiempo de cambio de formatos y/o referencia entre trabajos consecutivos. Al máximo seleccionado se le adiciona el tiempo de procesamiento y arranque de dicho trabajo en la máquina 1. Este caso es mutuamente excluyente con el caso 4, si existe cambio de formato y/o referencia el caso 3 invalida a este último. Para tal fin, se dispone del conjunto de restricciones (16) que identifica para los trabajos programados consecutivamente si se requiere cambio de referencia y/o formato.
4	K | J | $temp\_camb\_l_{jj}$ | $ML=1$ | $ML=2$ | $ML=3$ | $ML=4$ 1 | 2 | - 2 | 3 | 240 3 | 1 | 10 4 | 4 | 0	Los conjuntos de restricciones (20), (21) y (22) corresponden al mismo caso 3 con la diferencia de que los trabajos programados consecutivamente no requieren cambio de referencia y/o formato. El tiempo de finalización para este caso, depende del máximo entre el tiempo en que está disponible el trabajo *j* y la diferencia entre el tiempo de finalización y arranque en la máquina uno del trabajo asignado a la posición anterior $k - 1$ (tiempo en que queda disponible la máquina 1), más el tiempo de procesamiento de dicho trabajo en la máquina 1.

Tabla 4 Tiempos de finalización de los trabajos programados en líneas

$$\sum_{j\ in\ J} on\_h_h\ \ y_{jkh} \leq 1 \qquad \forall\, k \in K, h \in H \qquad (23)$$

$$\sum_{k\ in\ K} on\_h_h\ \ y_{jkh} \leq 1 \qquad \forall\, j \in J, h \in H \qquad (24)$$

$$\sum_{k\ in\ K, h\ in\ H} on\_h_h\ \ y_{jkh} = 1 \qquad \forall\, j \in J \qquad (25)$$

$$on\_h_h\ \ y_{jkh} \leq elegi\_h_{jh}\ \ on\_h_h \qquad \forall\, j \in J, k \in K, h \in H \qquad (26)$$

$$\sum_{j\ in\ J} on\_h_h\ y_{jkh} \geq \sum_{j\ in\ J} on\_h_h\ y_{j(k+1)h} \qquad \forall\ k \in K, h \in H, k < |K| \qquad (27)$$

$$ini\_k1\_mh_{jh} \geq fin\_l_{j4l} - \sum_{k\ in\ K} procesam\_l_{jl}\ x_{jkl} + \sum_{k\ in\ K} arr\_ml\_l_{j4l}\ cp\_carros_{jl}\ x_{jkl} \qquad \forall\ j \in J, h \in H, l \in L \qquad (28)$$

$$ini\_k1\_mh_{jh} \geq \sum_{k\ in\ K} to\_h_h\ y_{jkh} \qquad \forall j \in J, h \in H \qquad (29)$$

$$fin\_h_{j1h} \geq ini\_k1\_mh_{jh} + arr\_mh\_h_{j1h}\ y_{j1h} + procesam\_h_{jh}\ y_{j1h} \qquad \forall j \in J, h \in H \qquad (30)$$

$$fin\_h_{j,mh,h} \geq fin\_h_{j(mh-1)h} + \sum_{k\ in\ K} arr\_mh\_h_{j,mh,h}\ y_{jkh} \qquad \forall\ j \in J, h \in H, mh \in MH,\ mh > 1 \qquad (31)$$

$$y_{jkh} + y_{i(k+1)h} \leq cambio\_real\_h_{jih} + 1 \qquad \forall\ k \in K, j \in J, i \in J, h \in H\ k < |K|, i <>\text{j},\ posibles\_cambios\_h_{ji} = 1 \qquad (32)$$

$$ini\_k2\_mh_{jh} + M(1 - y_{jkh}) \geq fin\_l_{j4l} - procesam\_l_{j,l} + arr\_ml\_l_{j4l}\ cp\_carros_{jl} - M(1 - y_{i(k-1)h}) - M\left(1 - \sum_{w\ in\ K} x_{jwl}\right) \qquad \forall\ i \in J, h \in H, l \in L, j \in J,\ k \in K,\ i <> j, k > 1 \qquad (33)$$

$$ini\_k2\_mh_{jh} + M(1 - y_{jkh}) \geq fin\_h_{i1h} - arr\_mh\_h_{i1h} - M(1 - y_{i(k-1)h}) - M(cambio\_real\_h_{ijh}) \qquad \forall\ i \in J, h \in H, j \in J, k \in K,\ i <> j, k > 1 \qquad (34)$$

$$ini\_k2\_mh_{jh} + M(1 - y_{jkh}) \geq fin\_h_{i3h} + temp\_camb\_h_{ij} - M(1 - y_{i(k-1)h}) - M(1 - cambio\_real\_h_{ijh}) \qquad \forall\ i \in J, h \in H, j \in J, k \in K,\ i <> j, k > 1 \qquad (35)$$

$$fin\_h_{j1h} + M(1 - y_{jkh}) \geq ini\_k2\_mh_{jh} + procesam\_total_{j1h} \qquad \forall\ h \in H, j \in J, k \in K,\ k > 1 \qquad (36)$$

$$procesam\_total_{j1h} + M(1 - y_{jkh}) \geq procesam\_h_{jh} + arr\_mh\_h_{j1h} - M(1 - y_{i(k-1)h}) - M(1 - cambio\_real\_h_{ijh}) \qquad \forall\ i \in J, k \in K, j \in J, h \in H,\ i <> j, k > 1 \qquad (37)$$

$$procesam\_total_{j1h} + M(1 - y_{jkh}) \geq procesam\_h_{jh} - M(1 - y_{i(k-1)h}) - M(cambio\_real\_h_{ijh}) \qquad \forall\ i \in J, k \in K, j \in J, h \in H,\ i <> j, k > 1 \qquad (38)$$

$$t_j, fin\_l_{j\ ml\ l}, ini\_k1\_m1_{jl}, ini\_k2\_m1\_0c_{jl}, ini\_k2\_m1_{jl}, fin\_h_{j\ mh\ h}, ini\_k1\_mh_{jh}, ini\_k2\_mh_{jh}, procesam\_total_{j\ mh\ h} \geq 0 \qquad \forall\ j \in J, ml \in ML, l \in L,\ mh \in MH\ h\ in\ H \qquad (39)$$

De la misma manera que sucede son las líneas, los cinco primeros conjuntos de restricciones del FS de hornos (23), (24), (25), (26), (27) garantizan una asignación factible mientras los restantes (conjunto 28 a 39) consideran los diferentes casos para el cálculo del tiempo de finalización de los trabajos a programar.

En cuanto a las restricciones para el cálculo del tiempo de finalización de los trabajos asignados a un horno h, se conservan los cuatro casos planteados en la Tabla 4 para el cálculo de este, contemplando las siguientes variaciones.

- Caso 1: Los conjuntos de restricciones (28), (29) y (30) hacen alusión al caso en el cual el tiempo de finalización del trabajo j depende del máximo entre el tiempo en que está disponible el horno h y el tiempo en que queda disponible el primer carro del PFS de líneas, más el tiempo de procesamiento y arranque de dicho trabajo j en la máquina 1.
- Caso 2: El conjunto de restricciones que hace referencia al caso 2 es el (31).
- Casos 3 y 4: La evaluación de ambos casos se consideran de manera conjunta, conservando la misma lógica en que fueron calculados en líneas. Los conjuntos de restricciones (33), (34) y (35) identifican el máximo entre tres valores, el primero, la suma del tiempo de finalización en la última máquina del trabajo asignado a la posición anterior $k-1$ con el tiempo de cambio de formatos entre trabajos consecutivos, el segundo, el tiempo en que queda disponible la máquina 1 tras procesar el trabajo asignado a la posición inmediatamente anterior $k-1$ y el tercero, el tiempo en que es obtenido el primer carro de la línea l. Cabe resaltar que el funcionamiento de estas restricciones varía según el requerimiento de cambio de formato entre ordenes de fabricación consecutivas.

Posteriormente, el cálculo del tiempo de finalización del trabajo *j* para estos casos, se determina a partir de los conjuntos de restricciones (36), (37) y (38) que adicionan al máximo encontrado previamente el

tiempo de arranque y/o procesamiento de dicho trabajo en la máquina 1, dependiendo de si se requiere o no un cambio de formato entre ordenes de fabricación consecutivas.

Cabe añadir que el conjunto (32) es el encargado de identificar para los trabajos programados consecutivamente si se requiere o no cambio de formato.

Finalmente, como complemento al modelo y considerando que el objetivo del caso de estudio es minimizar la tardanza total, teniendo en cuenta la importancia del producto y del cliente, al modelo expuesto se le aplican dos variaciones, en las cuales la función objetivo para cada una tiene un enfoque diferente, como se muestra a continuación (Conjunto de funciones objetivo (40) y (41)).

$$\min z = \sum_{j\ in\ J} \sum_{c\ in\ C} t_j * cliente_{cj} * w\_cliente_c \quad (40)$$

$$\min z = \sum_{j\ in\ J} \sum_{c\ in\ C} t_j * producto_{pj} * w\_producto_p \quad (41)$$

Las funciones objetivo planteadas en las ecuaciones (40) y (41) están orientadas a minimizar la tardanza total ponderada con base en la prioridad de los clientes y productos, respectivamente.

4.2. Fase 2 - Metaheurística

Una vez realizada la programación de producción utilizando el modelo matemático se diseñó la técnica de solución en Visual Basic for Applications de Microsoft Excel integrando la metaheurística GRASP, el algoritmo PAES y la metodología AHP. A continuación, se explica cómo se aplica cada una de las anteriores herramientas para la construcción de la técnica diseñada.

4.2.1. Metaheurística GRASP

GRASP es una metaheurística que consta de dos fases en cada una de sus iteraciones: la fase constructiva y la fase de búsqueda local. En primer lugar, en la fase constructiva se incluyen aleatoriamente órdenes de producción a la programación con base en dos aspectos importantes, la función de utilidad y la Lista Restringida de Candidatos (RCL por sus siglas en inglés). En el primer aspecto, la función de utilidad implementada en la programación se basa en la regla de despacho Modified Due Date (MDD) (Baker & Bertrand, 1982).

Esta regla de despacho es una combinación de las reglas Earliest Due Date (EDD) y Shortest Processing Time (SPT) que intenta capturar los beneficios de cada una de ellas (Geiger & Uzsoy, 2006). Esta regla busca programar en primera instancia los trabajos que tienen un menor lapso para ser entregados, atendiendo el tiempo en que se encuentra el sistema, y en último lugar aquellos que tienen fechas de entrega lejanas, sin importar si el trabajo se encuentra tarde o no. Sin embargo, al realizar la experimentación correspondiente al apartado 5.2.1. se evidencia que la cantidad de iteraciones propuesta no es suficiente, por lo tanto, se requiere de un mayor tiempo de ejecución para encontrar soluciones más cercanas a la óptima. Adicionalmente, para la organización no es conveniente disponer de tiempos de ejecución superiores a los evidenciados con 1000 iteraciones. Con base en lo anterior, se modifica y restringe la función de utilidad (ecuación (42)):

$$f(x) = \begin{cases} Si \max\left(d_j - \max(t_l + p_{jl} + p_{jh}\ ,\ t_h + p_{jh}); 0\right) = 0 & M \\ dlc & d_j - t \end{cases} \quad (42)$$

Donde d_j es la fecha de entrega del trabajo j, t_l y t_h el tiempo en que la línea l y el horno h quedan disponibles para programar un nuevo trabajo, p_{jl} y p_{jh} el tiempo de procesamiento del trabajo j en la línea l y el horno h respectivamente, y t el tiempo en que queda disponible la línea o el horno (t_l o t_h). Con dicha función de utilidad se busca programar en primera instancia los trabajos que tienen un menor lapso para ser entregados y alcanzan a ser procesados tanto en líneas como en hornos atendiendo su disponibilidad. Aquellos trabajos que están tarde toman un valor M, correspondiente a número muy grande, que causa que estos sean programados en últimas posiciones.

Por otro lado, el segundo aspecto importante de la fase constructiva corresponde a la RCL la cual se compone de los trabajos que pueden ser incorporados paso a paso a la solución parcial en construcción, sin destruir la factibilidad de la solución y disminuyendo los costos (Gendreau & Potvin, 2010). Las órdenes de producción que ingresan a la RCL en la herramienta diseñada cumplen la siguiente condición (ecuación 43):

$$RCL = \{x | L \leq f(x) \leq L + \alpha(U - L)\} \quad (43)$$

Donde $f(x)$ es la función de utilidad del elemento x, α es un número entre 0 y 1, L es el menor valor de la función de utilidad encontrado y U es el mayor. Una vez calculada la función de utilidad se seleccionan las combinaciones línea-horno que cumplen el rango establecido en la ecuación 43 y aleatoriamente se elige una de ellas para ser programada. Tras ser programado el trabajo se actualizan los tiempos actuales de las líneas t_l y hornos t_h. Este proceso debe realizarse hasta que todos los trabajos hayan sido asignados. En la ilustración 5 se muestra el diagrama de flujo que representa la fase constructiva de la metaheurística.

En segundo lugar, la fase de búsqueda local examina la vecindad de la solución actual hasta encontrar un mínimo local (Gendreau & Potvin, 2010) optimizando la programación de los PFS de hornos. En el presente caso se realizan intercambios entre pares de trabajos de la solución arrojada por la fase constructiva. Una vez hecho un intercambio se calcula la tardanza total, si la nueva programación es factible y minimiza la función objetivo se guarda como la solución actual. Se deben evaluar todos los intercambios posibles, pero se guarda la programación que genere una mayor minimización a partir del movimiento entre dos trabajos, esta regla es conocida como Best-Improvement. En la ilustración 6 se muestra el diagrama de flujo que representa la fase de búsqueda local implementada en la técnica diseñada. Por su parte, en la ilustración 7 se evidencia el diseño completo de la metaheurística GRASP que integra tanto la fase constructiva como de búsqueda local analizadas anteriormente.

4.2.2. Algoritmo PAES

Es un algoritmo caracterizado por la optimización multiobjetivo que no se restringe a la búsqueda de una única solución, sino a un conjunto de soluciones llamadas soluciones *No-Dominadas*. Cada solución de este conjunto se dice que es un Óptimo de Pareto y, al representarlas en el espacio de los valores de las funciones objetivo, conforman lo que se conoce como Frontera de Pareto o Archivo de Pareto (Knowles & Corne, 1999).

El algoritmo PAES se integró al diseño actual al momento de considerar los criterios cualitativos: importancia del cliente e importancia del producto, con el fin de establecer las alternativas (soluciones de la frontera de Pareto) que mejor convengan a la empresa y que, finalmente, son evaluadas en la metodología AHP. Para implementar este algoritmo se debe enfocar cada iteración GRASP de la técnica diseñada a un objetivo diferente, así, la iteración 1 se orienta a la minimización de la tardanza total (objetivo 1), la iteración 2 a la minimización de la tardanza total ponderada atendiendo la importancia del cliente (objetivo 2), la iteración 3 a la minimización de la tardanza total ponderada considerando la importancia del producto (objetivo 3), la iteración 4, de nuevo, a la minimización de la tardanza total y así sucesivamente. El enfoque que se da a cada iteración GRASP se genera al seleccionar aleatoriamente de la RCL el trabajo que tenga mayor prioridad de acuerdo con la función objetivo que se trabaje en dicha iteración. Por ejemplo, en el caso de que en la RCL haya trabajos de dos tipos de clientes y el enfoque de la iteración sea minimizar la tardanza total ponderada atendiendo la importancia del cliente, el aleatorio se realiza, en primera instancia, entre los trabajos con el tipo de cliente de mayor prioridad. Posteriormente, al final de cada iteración se evalúa la solución de acuerdo con los tres objetivos antes mencionados equivalentes a las ecuaciones (44), (45) y (46). Para evaluar los objetivos en mención, el peso que tiene cada uno de los clientes y productos para la empresa se detalla en la Tabla 5.

$$Objetivo\ 1 = Tardanza\ Total\ (TT) \tag{44}$$

$$Objetivo\ 2 = 0{,}6\ TT_{Sodimac} + 0{,}1\ TT_{Exportación} + 0{,}3\ TT_{Nacional} \tag{45}$$

$$Objetivo\ 3 = 0{,}4\ TT_{p1} + 0{,}32\ TT_{p2} + 0{,}1\ TT_{p3} + 0{,}08\ TT_{p4} + 0{,}06\ TT_{p5} + 0{,}04\ TT_{p6} \tag{46}$$

Importancia cliente		**Importancia producto**					
Sodimac	60%	**Prioridad 1**	40%	Nuevo	**Prioridad 4**	8%	En desarrollo
Exportación	10%	**Prioridad 2**	32%	Activo	**Prioridad 5**	6%	Portafolio ofertas
Nacional	30%	**Prioridad 3**	10%	Para licitación Canal constructor	**Prioridad 6**	4%	Otros

Tabla 5. Importancia de los criterios cualitativos para la empresa.

Una vez es evaluada la solución de cada iteración, se debe comparar con aquellas que están incluidas en el *Archivo de Pareto*, si alguna o algunas están *dominadas* por la solución actual deben ser eliminadas y esta debe ser agregada al archivo, en caso contrario, debe descartarse. Al final de las iteraciones se cuenta con el *Archivo de Pareto* que contiene las secuencias de programación con mejores resultados. En la ilustración 8, se muestra el diagrama de flujo que representa el algoritmo PAES para la técnica diseñada.

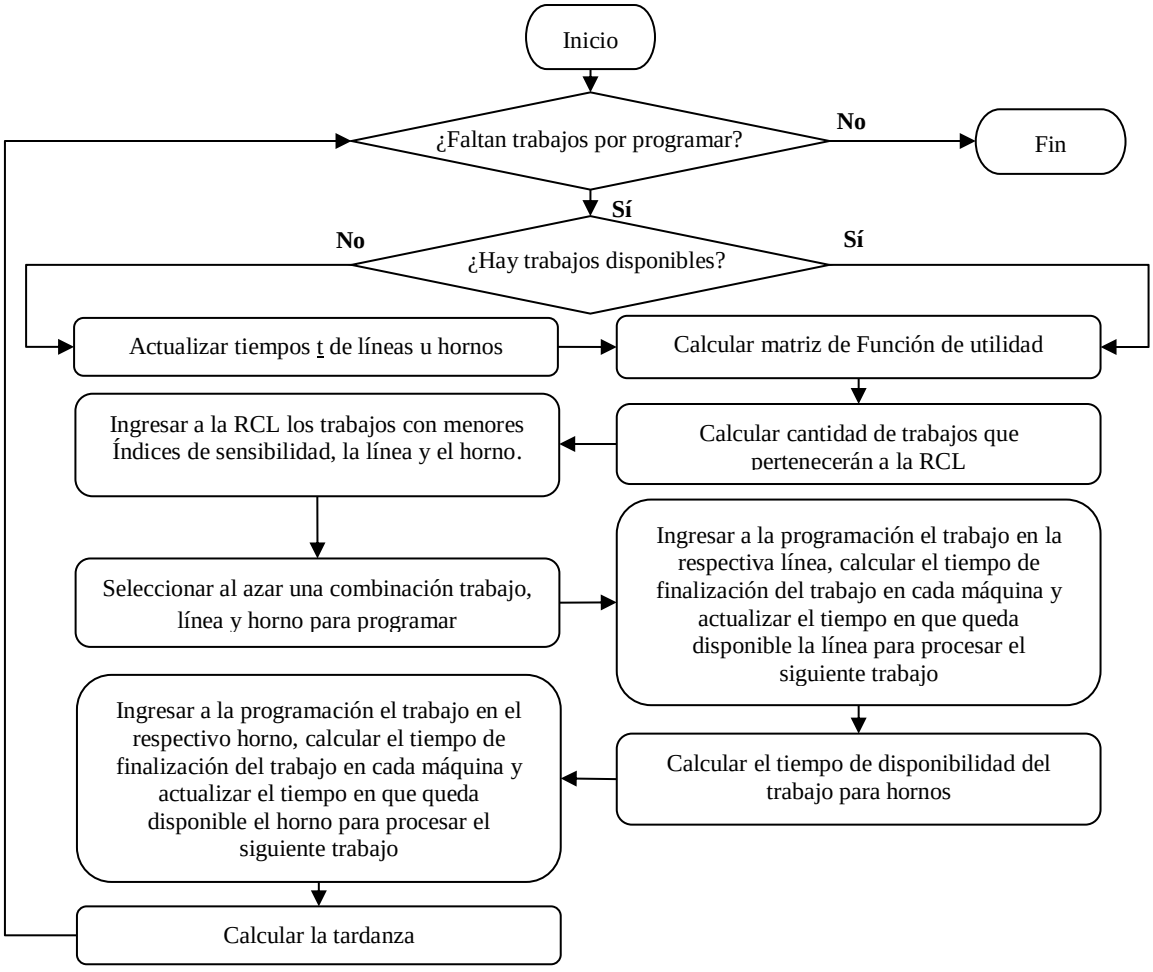


Ilustración 5. Fase constructiva metaheurística GRASP

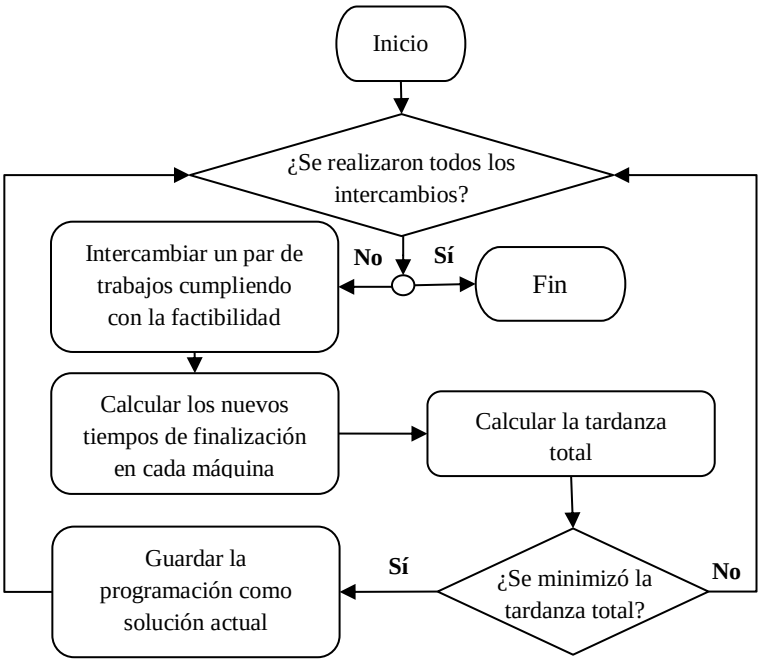


Ilustración 6. Búsqueda Local Metaheurística GRASP

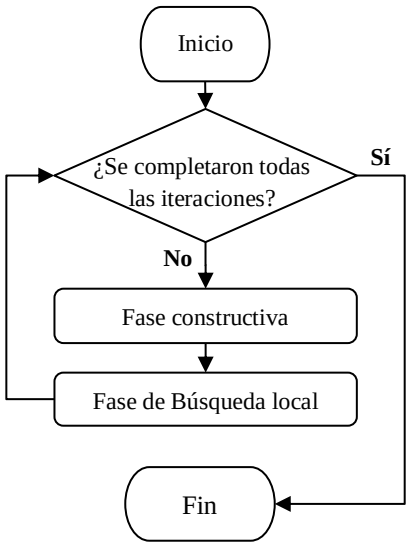


Ilustración 7. Metaheurística GRASP

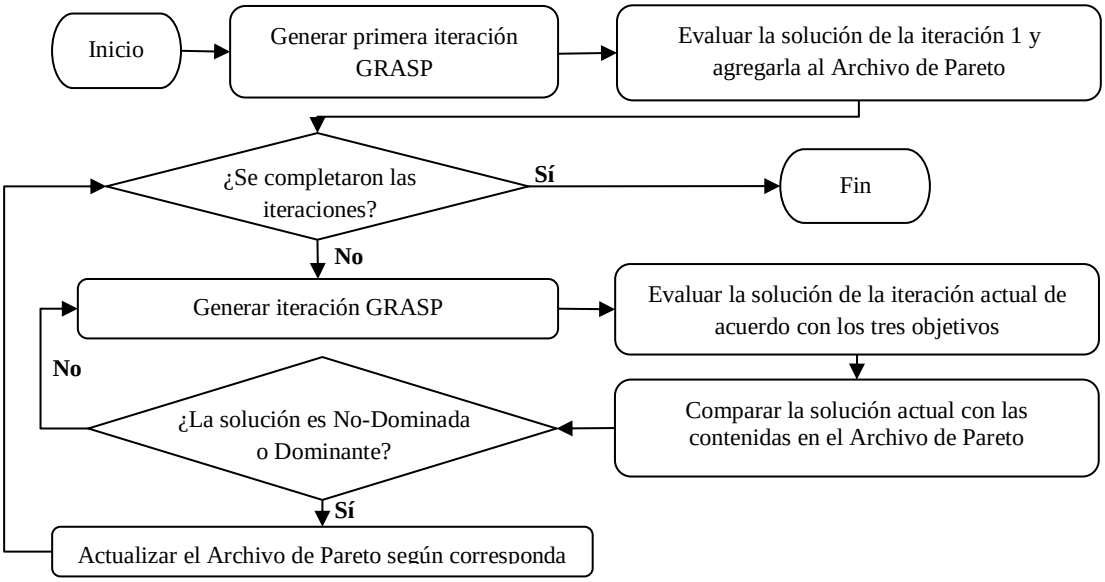


Ilustración 8. Algoritmo PAES

4.2.3. Metodología AHP

Es una metodología que utiliza escalas de comparación entre elementos, construyendo matrices y usando elementos del álgebra matricial para establecer prioridades (Saaty, 1990). A partir de las soluciones contenidas en el *Archivo de Pareto* se implementa el método AHP (ilustración 9) para determinar cuál es la solución que más se ajusta a los requerimientos de la empresa contemplando la ponderación de los tres criterios. En primer lugar, se calculan cuatro *matrices de escala de comparación pareada*, una entre criterios como se muestra en la Tabla 6, y tres entre alternativas, cada una de estas últimas orientada a un criterio específico. Posteriormente, se calcula para cada matriz un *vector propio*, para la primera de ellas este representa una priorización entre los criterios como se muestra en la Tabla 6, y para las demás, representa una priorización entre alternativas según cada criterio. Finalmente, los

vectores propios orientados a cada criterio deben ser multiplicados por el vector propio general para obtener cual es la alternativa que más conviene a la empresa según sus requerimientos.

Escala de comparación pareada entre criterios				Vector propio
Criterios	**Tardanza**	**Tardanza total ponderada atendiendo la Importancia del cliente**	**Tardanza total ponderada atendiendo la Importancia del producto**	
Tardanza Total	1	1/5	7	0,2271
Tardanza total ponderada atendiendo la Importancia del cliente	5	1	9	0,7219
Tardanza total ponderada atendiendo la Importancia del producto	1/7	1/9	1	0,0510

Tabla 6. Matriz de escala de comparación pareada entre criterios y Vector propio.

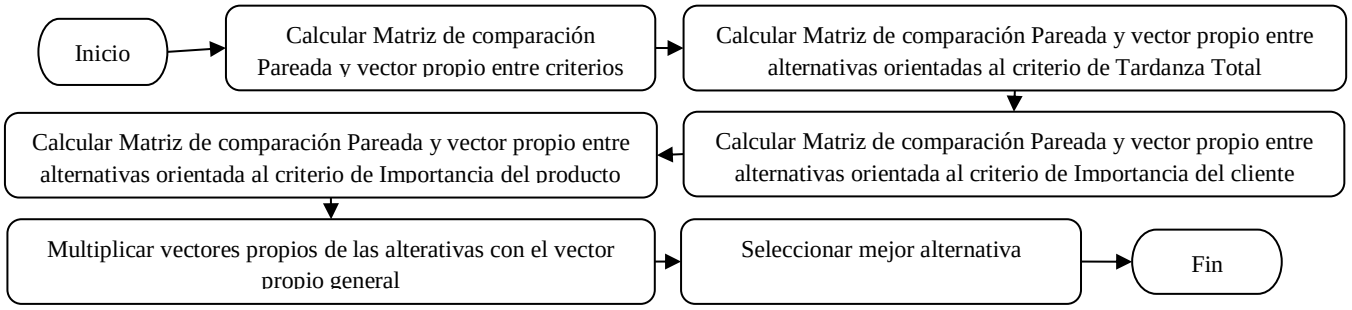


Ilustración 9. Metodología AHP

En la ilustración 10, se muestra el pseudocódigo de la técnica diseñada.

```
PROCEDIMIENTO Metaheurística GRASP ()

Declarar variables
Leer parámetros
Total_iteraciones = M
it=1
Cont_iteraciones = 1
Sum_trabajos=cantidad de trabajos a programar
MIENTRAS Cont_iteraciones <= M
        SI it=1
        'Programación orientada a: Tardanza Total
        'Programación Línea-Horno
        MIENTRAS Sum_trabajos<>0
                Fase constructiva combinación línea-horno

                Sum_trabajos= Sum_trabajos − 1
        FIN MIENTRAS
        Búsqueda Local
        FIN SI
        SI it=2
        'Programación orientada a: Tardanza Total enfocado a la importancia del cliente
        FIN SI
        SI it=3
        'Programación orientada a: Tardanza Total enfocado a la importancia del producto
        FIN SI
        SI it=3
        'Programación orientada a: Tardanza Total enfocado a la importancia del producto
        FIN SI
        'Cambio de iteración
        SI it=1
                it=2
        SI NO
                SI it=2
                        it=3
                SI NO
                        it=1
                FIN SI
        FIN SI
        'Archivo de pareto
        Algoritmo PAES
FIN MIENTRAS
'Selección de mejor alternativa
Metodología AHP
Imprimir

FIN PROCEDIMIENTO
```

Ilustración 10. Pseudocódigo de la metaheurística diseñada

4.3. Fase 3 – Regla de despacho ATCS

Para medir la calidad de la metaheurística propuesta se implementa la regla de despacho ATCS como marco de comparación. La ATCS busca minimizar la tardanza total ponderada cuando los trabajos están sujetos a tiempos de alistamiento. Esta regla la integran tres etapas, la primera consiste en un análisis estadístico de la instancia del problema, la segunda, construye una programación basándose en la ecuación (51), la cual contiene dos parámetros, $\boldsymbol{k_1}$ y $\boldsymbol{k_2}$, resultantes de la fase anterior y, la tercera, un procedimiento de mejoramiento de la solución obtenida en la segunda fase (Lee et al., 1997).

4.3.1. Etapa 1 - Análisis estadístico

En esta etapa tres coeficientes son calculados para caracterizar la instancia como se muestra en la tabla 7 (ecuación 47 a 49). Adicionalmente, para estimar el valor del $\boldsymbol{C_{max}}$ se plantea la ecuación (50), en donde $\boldsymbol{n}$ es la cantidad de trabajos a programar, $\boldsymbol{L}$ y $\boldsymbol{H}$ la cantidad de líneas y hornos que participan en la programación y $\boldsymbol{\beta}$ es un factor menor o igual a $\mathbf{1}$ que se ve afectado por la variabilidad de los tiempos de alistamiento. En el presenta caso, dado que los tiempos de alistamiento entre trabajos no varían el $\boldsymbol{\beta}$ se considera igual a 1.

Nombre	Formula	Descripción	Ecuación
Due Date Tightness factor (τ)	$\tau = 1 - \frac{\bar{d}}{C_{max}}$	Donde $\bar{d}$ es el promedio de las fechas de entrega y C_{max} el makespan (el tiempo de finalización del último trabajo programado).	(47)
Due Date Range factor (R)	$R = \frac{d_{max} - d_{min}}{C_{max}}$	Donde d_{max} y d_{min} corresponden al mayor y menor de las fechas de entrega.	(48)
Setup Time Severity factor (η)	$\eta = \frac{\bar{s}}{\bar{p}}$	Donde $\bar{s}$ y $\bar{p}$ hacen referencia al tiempo promedio de alistamiento y procesamiento.	(49)

Tabla 7. Coeficientes etapa 1 ATCS

$$C_{max} = \frac{n(\bar{\bar{p}} + \beta\bar{s})}{\max(L; H)} \quad (50)$$

4.3.2. Etapa 2 - Regla ATCS

El índice de prioridad $(\boldsymbol{I_j})$ de la regla de despacho ATCS se define según la ecuación (51):

$$I_j(t,u) = \frac{w_j}{p_j}\exp\left(-\frac{\max\left(d_j - p_j - \max(t_l, t_h), 0\right)}{k_1\bar{p}}\right)\exp\left(-\frac{s_{uj}}{k_2\bar{s}}\right) \quad (51)$$

Donde $\boldsymbol{w_j}$ es el peso de cada trabajo, $\boldsymbol{t_l}$ y $\boldsymbol{t_h}$ representan el tiempo en que la línea $\boldsymbol{l}$ y el horno $\boldsymbol{h}$ quedan disponibles para programar un nuevo trabajo, $\boldsymbol{u}$ es el trabajo que acaba de completarse y $\boldsymbol{k_1}$ y $\boldsymbol{k_2}$ son parámetros de escala calculados conforme a las ecuaciones (52) y (53).

$$\begin{matrix} k_1 = 4.5 + R & Si\ R \leq 0.5 \\ k_1 = 6.9 - 2R & Si\ R \geq 0.5 \end{matrix} \quad (52) \qquad k_2 = \frac{\tau}{2\sqrt{n}} \quad (53)$$

Cabe añadir que para determinar $\boldsymbol{w_j}$ se calcula, en primer lugar, una matriz de escala de comparación pareada entre los dos criterios, cualitativos obteniendo un peso del 90% para la importancia del cliente y del 10% para la importancia del producto. En segundo lugar, se suma el producto entre el porcentaje de importancia de cada criterio y el peso del nivel correspondiente.

Para realizar la programación se recalcula el índice $\boldsymbol{I_j}$ hasta que los trabajos hayan sido asignados en su totalidad. Tras cada actualización de los índices $\boldsymbol{I_j}$ el trabajo a asignar corresponde al de mayor valor.

4.3.3. Etapa 3 - Procedimiento de mejora

Para intentar mejorar la secuencia obtenida en la fase dos en términos de tardanza total ponderada, se aplica la misma metodología Best-Improvement implementada en la búsqueda local de la metaheurística. En la ilustración 11 se resume la regla de despacho ATCS.

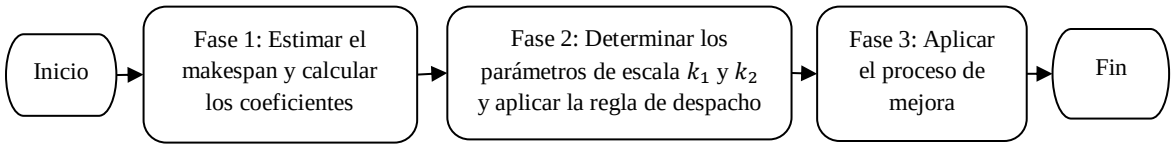


Ilustración 11. Etapas de la regla de despacho ATCS

5. Resultados

5.1. Modelo Matemático

Con el fin de medir la efectividad de la metaheurística en primera instancia se intentan comparar los resultados de la metaheurística con los del modelo matemático, programando las instancias que utilizó la empresa durante el trimestre de febrero a abril del 2018. Se conoce que el problema a resolver es de complejidad NP-hard, por consiguiente, el tiempo computacional que requiere el modelo matemático es elevado. Experimentalmente se demuestra que no es posible ejecutar instancias con más de 5 trabajos, dado que esto implicaría más de 24 horas de ejecución para una instancia de tan solo 6 trabajos, con un equipo de procesador Intel (R) CORE (TM) i7 octava generación, velocidad de 1,80 GHz Y RAM de 8 GB.

Con base en lo anterior, en la Tabla 8 se compara a escala ambas herramientas de programación, evaluando 10 instancias diferentes, cada una con 5 trabajos y con el objetivo de minimizar la tardanza total. Es necesario mencionar que en capítulos posteriores se parametriza la metaheurística, obteniendo que para instancias con menos de 22 trabajos se debe manejar un criterio de parada de 230 iteraciones y utilizar un RCL de 0,3.

Con base en los resultados obtenidos se construye la Tabla 8 en la cual se aprecia un tiempo promedio de ejecución de la metaheurística, atendiendo el criterio de parada, de 331 segundos, en los cuales en promedio a la doceava iteración se encuentra la mejor respuesta, equivalente a un tiempo promedio de 18 segundos. De lo anterior se concluye que el tiempo de obtención de la mejor respuesta encontrada con el modelo matemático es en promedio 160 veces mayor que el de la metaheurística. Por otro lado, se observa que no existe variación porcentual de la función objetivo entre el GRASP y el MILP para todas las instancias, mostrando que la metaheurística propuesta para pequeñas instancias alcanza la solución óptima.

DJ	1	2	3	4	5	6	7	8	9	10	Prom.
Tardanza GRASP y MILP (min)	882	870	897	450	645	1177	903	211	614	962	-
Tiempo MILP (seg)	4204	6053	7437	780	1299	3866	2187	963	1418	1300	2951
Tiempo GRASP 230 it. (seg)	322	335	341	319	310	388	325	313	317	339	331
Tiempo mejor solución GRASP (seg)	6	9	16	21	13	72	4	3	32	9	18

Tabla 8 Niveles del factor Cantidad de Iteraciones para cada DOE

5.2. Metaheurística

Con el fin de encontrar una de las mejores soluciones proporcionadas por la metaheurística, se procede a su parametrización. Para ello se propone encontrar los mejores valores para la cantidad de iteraciones que definen el criterio de parada de la herramienta y el α de la RCL.

5.2.1. Parametrización de la Metaheurística

La parametrización de la metaheurística se realiza con base en dos diseños de experimentos (DOE), explicados en capítulos posteriores, los cuales se desarrollan en torno a 6 instancias diferentes. En éstas varía la fecha de entrega de las ordenes de fabricación, se conserva la misma cantidad de líneas y hornos activos y se seleccionan 3 tamaños de instancias de acuerdo con la cantidad promedio de trabajos

procesados por la empresa diariamente (44 trabajos), y un menor (22 trabajos) y mayor valor (66 trabajos) a esta cantidad.

Las líneas y hornos activos en el DOE se seleccionan de acuerdo con la maquinaria que la empresa tuvo en funcionamiento la mayor parte del tiempo, durante el cuatrimestre de enero a abril del 2018 (7 líneas y 7 hornos). Por otro lado, la generación de las fechas de entrega para cada instancia se realiza de manera aleatoria con base en la distribución uniforme entre $P\left(1-T-\frac{R}{2}\right)$ y $P\left(1-T+\frac{R}{2}\right)$, en donde T y R son los parámetros Tardiness Factor o Tighness Factor y Due Date Range (Vallada, Ruiz, & Minella, 2008), que toman los valores $T=\{0.4,0.6\}$ y $R=\{0.2,0.6,1\}$ respectivamente y P hace referencia al mínimo tiempo requerido para fabricar las ordenes de producción solicitadas, siendo este la sumatoria entre el menor tiempo de procesamiento de un trabajo en una de las líneas disponibles $(Min\ p_{jl})$ y el menor tiempo de procesamiento de todos los trabajos en los hornos activos $(\sum Min\ p_{jh})$; dividido en la cantidad de hornos en funcionamiento h, ecuación (54).

$$P = Min\ p_{jl} + \frac{\sum Min\ p_{jh}}{h} \tag{54}$$

5.2.1.1. DOE para definir la cantidad de iteraciones

Con el fin de determinar la cantidad de iteraciones idónea que define el criterio de parada de la herramienta con relación al tamaño de la instancia se plantea el primer diseño de experimentos.

Para cada tamaño de instancia se realiza un DOE, en el cual se evalúan diferentes niveles para el factor Cantidad de Iteraciones y 6 diferentes tipos de due dates d_j generados como se menciona en el numeral 5.2.1., tenidos en cuenta como un bloque del diseño. Para definir los niveles de cada DOE (Tabla 9) se ejecuta cada instancia con 1000 iteraciones y se identifican los puntos de variación, como se muestra en la ilustración 12.

DOE	Niveles del factor Cantidad de Iteraciones
22 trabajos	15, 30, 45, 60, 75, 230, 300, 500, 700
44 trabajos	50, 100, 175, 250, 400, 525, 600, 750, 1000
66 trabajos	25, 50, 75, 100, 200, 400, 650, 900, 1000

Tabla 9 Niveles del factor Cantidad de Iteraciones para cada DOE

Al ejecutar la experimentación y comprobar los supuestos de normalidad, varianza constante e independencia, se puede afirmar con un nivel de confianza del 95% que existe un efecto significativo de la cantidad de iteraciones sobre la tardanza total en las tres instancias, 22, 44 y 66 trabajos. Considerando lo anterior, para determinar los niveles de los factores que generan una diferencia significativa se aplica el método Least Significant Difference (LSD) de Fisher para todas las instancias, obteniendo las gráficas de diferencias de medias y agrupaciones de Fisher que se muestran en las ilustraciones 13, 14 y 15 respectivamente.

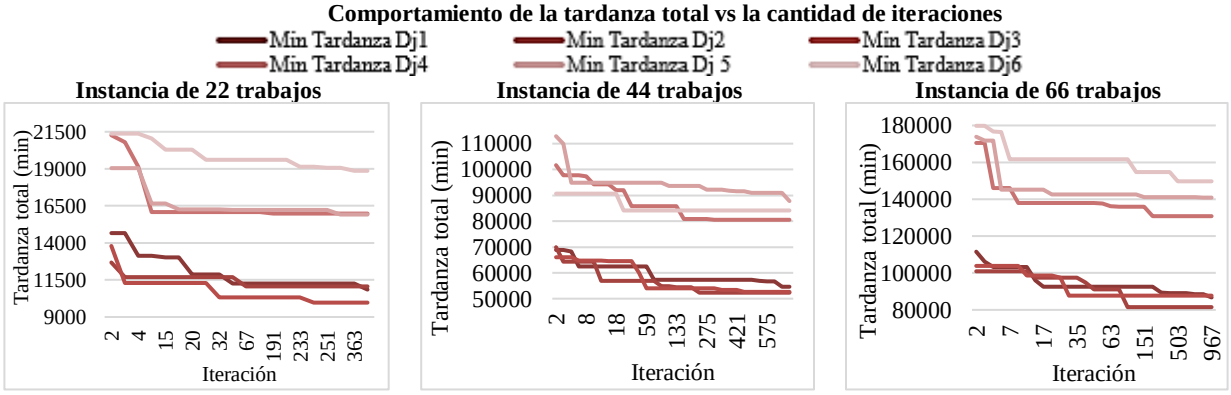
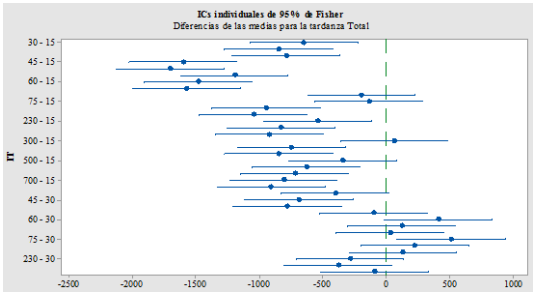
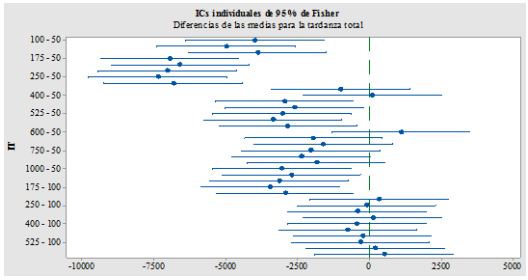


Ilustración 12. Comportamiento de la tardanza total Vs la cantidad de iteraciones para instancia de 22, 44 y 66 trabajos



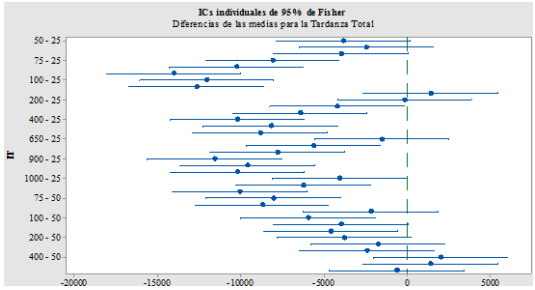
IT	N	Media	Agrupación			
15	6	16310,2	A			
30	6	15658,3		B		
60	6	15519,0		B	C	
45	6	15460,0		B	C	
300	6	15115,8			C	D
500	6	14829,3				D E
700	6	14737,1				D E
75	6	14711,0				D E
230	**6**	**14609,9**				**E**

lustración 13. Diferencia de medias para la tardanza total y Agrupación Fisher - Instancia de 22 trabajos (DOE 1)



IT	N	Media	Agrupación		
50	6	75492,1	A		
250	6	71602,9		B	
100	6	71502,6		B	
175	6	70505,4		B	C
525	6	68898,9			C
1000	6	68669,5			C
400	**6**	**68557,0**			**C**
600	6	68463,6			C
750	6	68145,6			C

lustración 14. Diferencia de medias para la tardanza total y Agrupación Fisher - Instancia de 44 trabajos (DOE 2)



IT	N	Media	Agrupación		
25	6	124679	A		
75	6	122221	A		
50	6	120838	A		
100	6	120687	A		
200	6	116603		B	
400	6	114419		B	C
900	6	112634		B	C
1000	6	112007			C
650	**6**	**110642**			**C**

lustración 15. Diferencia de medias para la tardanza total y Agrupación Fisher - Instancia de 66 trabajos (DOE 3)

Según la información anterior, si un intervalo no contiene cero o si las medias no comparten una letra, son significativamente diferentes, por lo tanto, la cantidad de iteraciones para una instancia de 22 trabajos que generan una diferencia significativa son 15, 30 y 230, sin embargo, se escoge 230 correspondiente a la menor media. Paralelo a esto, para las instancias de 44 y 66 trabajos se seleccionan 400 y 650 iteraciones, las cuales hacen referencia a la menor cantidad de iteraciones que pertenecen al grupo con menor media.

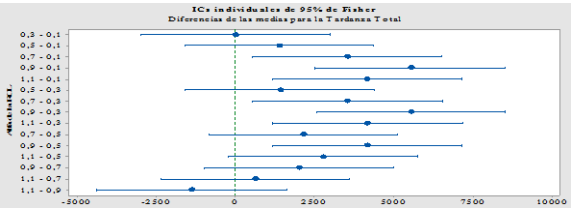
5.2.1.2. DOE para definir el α de la RCL

Una vez se ha determinado la cantidad de iteraciones con las cuales debe ejecutarse cada una de las instancias evaluadas, se procede a determinar el valor del α de acuerdo con la ecuación (43), de tal manera que se minimice la tardanza, variando el tamaño de la RCL. De modo que, si se toman valores pequeños, la proporción de combinaciones línea-horno que entran a la RCL con respecto a la totalidad es baja en comparación con valores de α altos.

Para determinar el valor de α se realizan tres DOE diferentes, uno para cada tamaño de instancia, en los cuales se consideran 6 diferentes niveles de α (0.1, 0.3, 0.5, 0.7, 0.9, 1.1) y 6 distintos d_j generados como se mencionó anteriormente y tenidos en cuenta como un bloque del diseño. Cabe aclarar que, la cantidad de iteraciones con las cuales se corre cada instancia depende de las cantidades seleccionadas en el numeral 5.2.1.1.

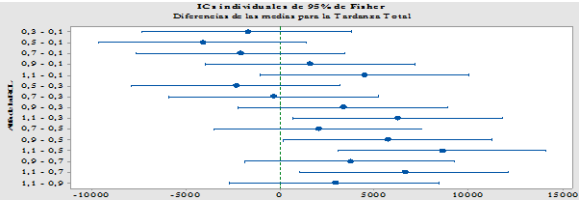
Al ejecutar la experimentación y comprobar los supuestos de normalidad, varianza constante e independencia, se puede afirmar con un nivel de confianza del 95% que existe un efecto significativo

del α sobre la tardanza total en las instancias de 44 y 66 trabajos. En el caso de 22 trabajos se obtuvo un valor-p de 0.717 lo cual revela que no se encontró efecto significativo. Considerando lo anterior, para determinar cuáles son los valores de α que generan una diferencia significativa se aplica el método LSD de Fisher para las instancias de 44 y 66 trabajos, obteniendo las gráficas de diferencias de medias y agrupaciones de Fisher expuestas en las ilustraciones 16 y 17.



α	N	Media	Agrupación		
0,9	6	73603,5	A		
1,1	6	72222,8	A	B	
0,7	6	71596,2	A	B	
0,5	6	69456,2		B	C
0,1	6	68078,2			C
0,3	**6**	**68060,2**			**C**

lustración 16. Diferencia de medias para la tardanza total y Agrupación Fisher - Instancia de 44 trabajos



α	N	Media	Agrupación		
1,1	6	119481	A		
0,9	6	116595	A	B	
0,1	6	115028	A	B	C
0,3	6	113268		B	C
0,7	6	112903		B	C
0,5	**6**	**110907**			**C**

lustración 17. Diferencia de medias para la tardanza total y Agrupación Fisher - Instancia de 66 trabajos

Según la información anterior, si un intervalo no contiene cero o si las medias no comparten una letra, son significativamente diferentes, por lo tanto, los niveles del α de la RCL para una instancia de 44 trabajos que generan una diferencia significativa son 0.1, 0.3 y 0.9, sin embargo, se escogió 0.3 correspondiente a la menor media; de igual forma, para una instancia de 66 trabajos se escogió un α de 0.5. Por otro lado, dado que para una instancia de 22 trabajos no existe efecto significativo se decide tomar un nivel de 0.3 al igual que la instancia de 44 trabajos.

5.2.2. Resultados de la Metaheurística GRASP

Teniendo en cuenta que se tiene parametrizada la metaheurística en términos de la cantidad de iteraciones y del α de la RCL, se procede a compararla con el escenario real de la empresa. Para realizar la analogía se tomaron los días en los cuales llegaron nuevas órdenes de producción durante el cuatrimestre de enero a abril del 2018 y se programaron en la herramienta diseñada de manera continua, es decir un día a la vez y de forma ordenada. Según lo anterior, es preciso decir que de la programación de un día se pueden obtener trabajos que deben ser reprogramados en el siguiente, esto dado que al analizar la fecha de inicio del trabajo en la planificación esta sobrepasa el tiempo de inicio del siguiente día a programar. Así mismo, para garantizar que la reprogramación se ejecute lo más cercano a la realidad, se debe calcular el tiempo en el cual quedan disponibles las líneas y los hornos día a día para iniciar la programación. De la misma manera que se determina si un trabajo debe ser reprogramado se calculan estos tiempos, teniendo en cuenta que si un trabajo empieza a fabricarse antes de que empiece el siguiente día del horizonte de planeación y termina en este último, el tiempo en que deje disponible la línea y el horno es el tiempo de inicio para cada uno de ellos en el siguiente día. Lo anterior se representa en la ilustración 18.

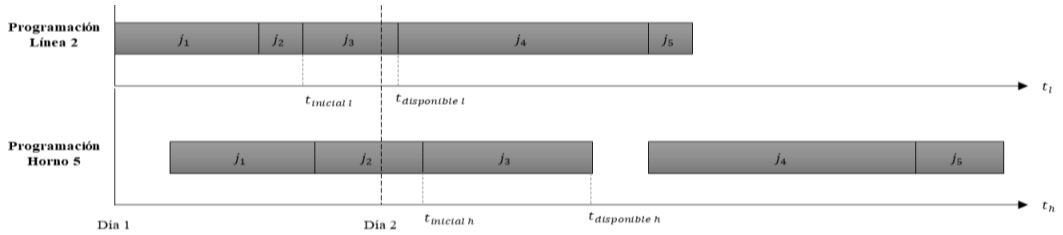


Ilustración 18. Ejemplo de reprogramación y de tiempos de inicialización

En el ejemplo de la ilustración 18, se muestra la programación de 5 trabajos realizada en el día 1. Dado que el trabajo 4 y 5 empezaron a ejecutarse después del día 2, deben reprogramarse para la planificación del siguiente día. Adicionalmente, debido a que el trabajo 3 empezó a ejecutarse en el día 1 y dejó disponible la línea en el día 2, se deben actualizar los tiempos de inicialización tanto de la línea 2 como del horno 5 para la programación del día 2, en este caso con el tiempo en que el trabajo 3 deja disponible la línea y el horno, $t_{disponible\ l}$ y $t_{disponible\ h}$ respectivamente.

Los resultados de la programación de la metaheurística considerando las órdenes de fabricación del escenario real de la empresa se muestran en la ilustración 19.

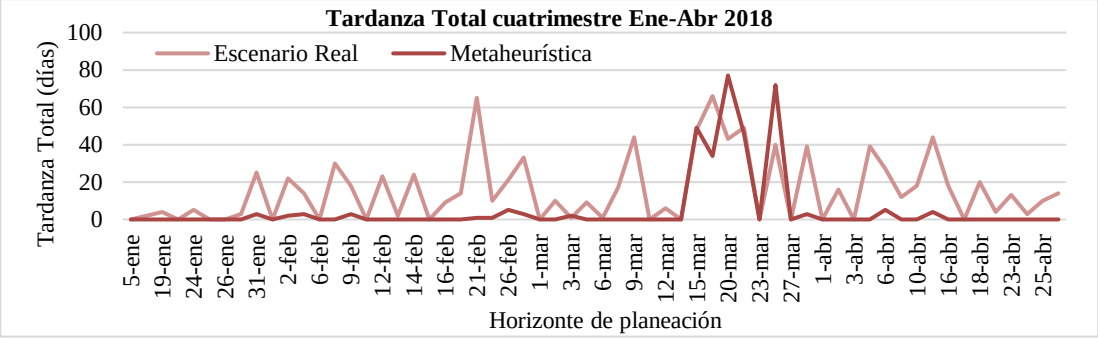


Ilustración 19. Tardanza Total cuatrimestre Ene-Abr de 2018 - Escenario real Vs Metaheurística

Se observa que la metaheurística genera un muy buen resultado en comparación con el modelo actual minimizando la tardanza de 935 días en el primer cuatrimestre del año a 313 días, lo que equivale a una reducción del 67% en la tardanza total. Con respecto al tiempo de ejecución un programa con 22 trabajos y 230 iteraciones dura en ejecución en promedio 42 minutos, con 44 trabajos y 400 iteraciones 72 minutos y con 66 trabajos y 650 iteraciones alrededor de 325 minutos.

5.2.3. Comparación entre funciones objetivo

Con el fin de evaluar el impacto de las tres funciones objetivo sobre la tardanza total se analizan 18 instancias de tres tamaños diferentes, 22, 44 y 66 respectivamente; cada una con 6 Dj distintos. En dichas instancias, como se muestra en la ilustración 20, al priorizar los trabajos de acuerdo con el tipo de cliente y el tipo de producto se evidencia un incremento sobre la tardanza, equivalente a 2 y 3 puntos porcentuales respectivamente, frente a una función objetivo que no incluye ningún criterio cualitativo.

Para corroborar el comportamiento identificado, se evalúan los resultados de la metaheurística para el cuatrimestre de enero a abril del 2018, con base en las tres funciones objetivo planteadas. Los resultados evidencian un incremento del 10% en la tardanza al priorizar los trabajos con base en los clientes más importantes y del 7% al priorizar los trabajos con base en los productos más significativos para la compañía. Con base en lo anterior, el impacto que genera la inclusión de los criterios cualitativos en la programación fluctúa de acuerdo con la variación de dichos criterios en las instancias y el peso asignado a cada uno de ellos.

5.2.4. Comparación con la Regla de Despacho ATCS

Con el fin de medir la calidad de la metaheurística se implementa la regla de despacho ATCS. En primer lugar, se comparan 18 instancias programadas a través de la regla de despacho con la metaheurística propuesta, 6 con cada tamaño de instancia (22, 44 y 66 trabajos) y cada una con un d_j diferente. En la ilustración 21 se muestran los resultados.

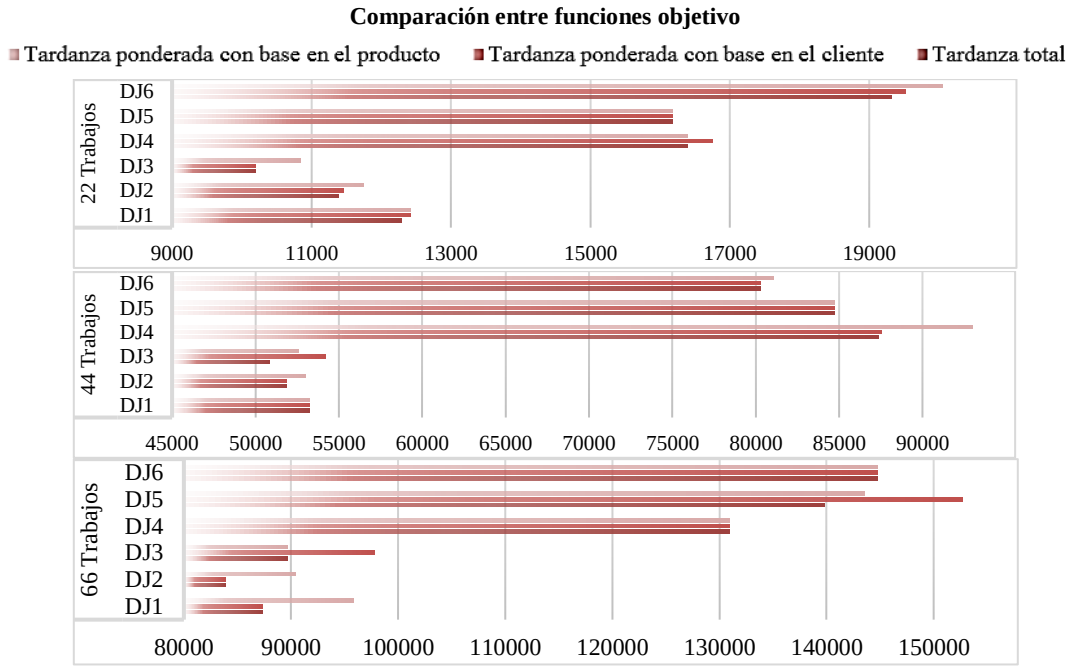


Ilustración 20. Comparación entre funciones objetivo para instancias de 22, 44 y 66 trabajos

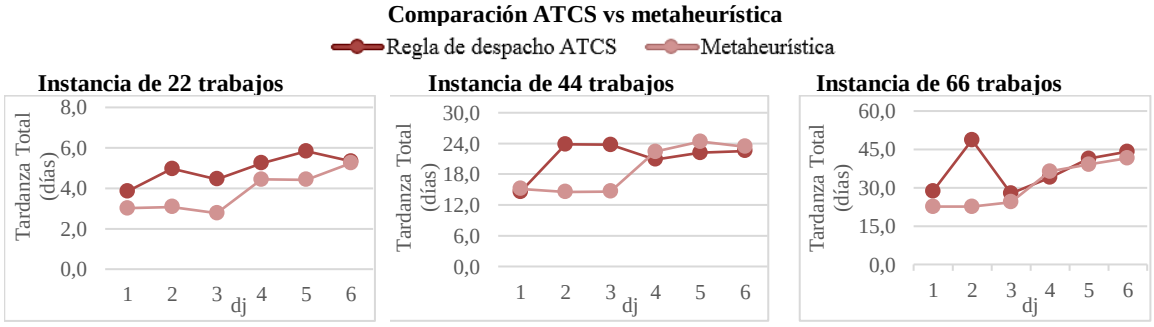


Ilustración 21. Comparación de 18 instancias entre regla despacho ATCS y metaheurística

Al comparar las 18 instancias, la metaheurística encuentra mejores resultados para el problema de la programación de la producción en comparación con la regla de despacho ATCS. El GRASP es un 23% mejor en comparación con el ATCS para una instancia de 22 trabajos, 10% mejor con respecto a una de 44 trabajos y 17% mejor para una de 66 trabajos. Para el conjunto de instancias evaluadas, la metaheurística presenta en promedio un mejor resultado, equivalente al 17%, evaluado en términos de la Tardanza Total.

En segundo lugar, así como se realizan las corridas de la metaheurística con base en el primer cuatrimestre del año se realiza el proceso de ejecución de la regla de despacho para este periodo, obteniendo los resultados mostrados en la ilustración 22.

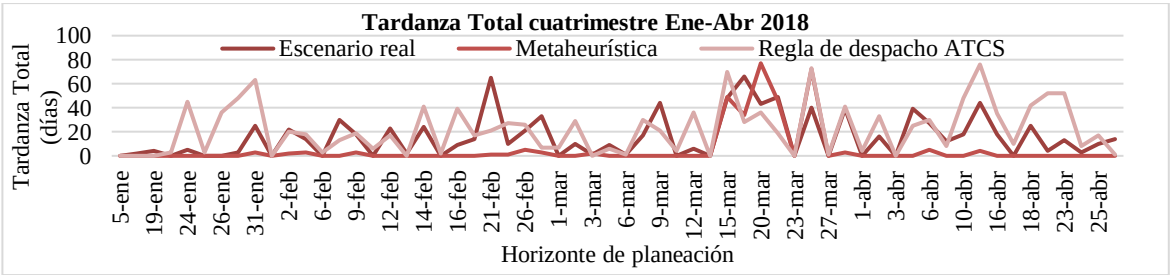


Ilustración 22. Tardanza Total cuatrimestre Ene-Abr de 2018 - Escenario real Vs Metaheurística Vs Regla de despacho

Se observa que la regla de despacho no genera un resultado superior en comparación con el escenario real y la metaheurística. A partir de los resultados obtenidos se concluye que la herramienta diseñada es superior en un 67% al modelo actual que desarrolla la empresa y en un 76% a la regla de despacho ATCS en términos de tardanza total; obteniendo este último 1317 días de tardanza en el cuatrimestre estudiado. Con respecto al tiempo de ejecución un programa con 22 trabajos tarda aproximadamente 1.6 minutos de ejecución, con 44 trabajos 6 minutos y con 66 alrededor de 13.4 minutos.

6. Conclusiones y recomendaciones

Este estudio está orientado al diseño de una metaheurística GRASP hibridizada con las metodologías PAES y AHP para la programación de la producción en un Distributed Permutation Flowshop with Line Elegibility (DPFSLE), minimizando la tardanza total, teniendo en cuenta criterios cualitativos, como: la importancia del cliente y del producto. Para demostrar la eficiencia de la metaheurística propuesta, se llevan a cabo un conjunto de diseños experimentales con el fin de parametrizar dicha herramienta. El desempeño de la metaheurística es superior en un 67% al modelo actual que desarrolla la empresa y en un 76% a la regla de despacho ATCS en términos de tardanza total. Además, se refleja un incremento del 8,5% en la tardanza al incluir los criterios cualitativos en la programación, con base en los trabajos programados en el escenario real de la empresa. Por otro lado, el tiempo de ejecución es en promedio 21 veces más lento que el de la regla de despacho ATCS debido a que es un proceso iterativo a diferencia de este último.

Los resultados indican para futuras extensiones a este diseño, que el lenguaje de programación Visual Basic For Applications puede ser reemplazado por uno más estructurado, organizado y flexible que le proporcione mayor velocidad de ejecución. De igual manera, el diseño puede ser complementado con la exploración de nuevas funciones de utilidad, estrategias de búsqueda local y reglas de despacho que permitan alcanzar mejores resultados.

7. Glosario

- **Baldosas:** Losa cerámica para pisos y paredes que posee una estructura parcialmente cristalina y vítrea. (Asociación Española de Técnicos Cerámicos, 2004)
- **Flow shop:** Entorno de producción en donde un conjunto de trabajos no relacionados debe ser procesados en un conjunto de máquinas. Estas máquinas se disponen en serie y cada trabajo tiene que visitar cada una de ellas en el mismo orden (Naderi & Ruiz, 2010).
- **Plan maestro de producción (MPS):** Es el cronograma de fabricación que contempla el pronóstico actualizado de demanda, el plan de producción aprobado, las estimaciones actuales de la disponibilidad de materiales y las estimaciones de la capacidad del sistema, así como los atrasos actuales de pedidos de clientes.(Ptak & Chad, 2011)
- **Programación de la producción:** Determinar cuándo se ejecutarán los pedidos de producción. El proceso consiste en determinar tiempos para la ejecución de las actividades de producción, luego conciliar el calendario con el plan de producción y finalmente apoyar las decisiones y acciones para lograr los objetivos de producción deseados. (Zandin, 2001)
- **Programación lineal:** Rama de la programación matemática que estudia problemas de optimización en los cuales se desea maximizar (o minimizar) una función lineal restringida mediante ecuaciones o desigualdades lineales. (Hernández, 2007)
- **Tiempo de alistamiento:** Los elementos del trabajo que comúnmente se incluyen en los estándares de preparación involucran a todos los eventos que ocurren entre la terminación de la tarea anterior y el inicio de la actual. Incluye elementos de “desarmado” y “guardado”, tomar los dibujos del despachador, preparar la máquina, retirar las herramientas de la máquina, regresar las herramientas a su depósito, contar la producción y/o alistar insumos. (Niebel & Freivalds, 2009)
- **Transfer:** Sistema de rieles que permiten el proceso de movilización entre las líneas de ensamble y los hornos. (Rodríguez, 2017)
- **Tiempo de arranque:** Tiempo en que es obtenida la primer baldosa en una máquina desde que ingresa a la misma.

Referencias

Asociación Española de Técnicos Cerámicos. (2004). *Tecnología cerámica aplicada - Volúmen 1*. IMPIVA, Diputació de Castelló, BANCAJA.

Asociación Española De Técnicos Cerámicos. (2001). *Tecnología cerámica aplicada - Volumen 2* (Vol. 2). IMPIVA, Diputació de Castelló, BANCAJA. https://doi.org/10.1073/pnas.0703993104

Azab, A., & Naderi, B. (2014a). Greedy heuristics for distributed job shop problems. *Procedia CIRP*, *20*(C), 7–12. https://doi.org/10.1016/j.procir.2014.05.025

Azab, A., & Naderi, B. (2014b). Modeling and heuristics for scheduling of distributed job shops. *Expert Systems with Applications*, *41*(17), 7754–7763. https://doi.org/10.1016/j.eswa.2014.06.023

Baker, K. R., & Bertrand, J. W. M. (1982). A Dynamic Priority Rule for Scheduling Against Due-Dates, *3*(November).

Bierwirth, & Kuhpfahl. (2017). Extended GRASP for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research*, *261*(3), 835–848. https://doi.org/10.1016/j.ejor.2017.03.030

Chang, H. C., & Liu, T. K. (2015). Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-015-1084-y

Chang, Lin, Pai, Zhong, & Hung. (2008). Ant colony optimization system for a multi-quantitative and qualitative objective job-shop parallel-machine-scheduling problem. *International Journal of Production Research*, *46*, 5719–5759.

Chang, P. T., & Lo, Y. T. (2001). Modelling of job-shop scheduling with multiple quantitative and qualitative objectives and a GA/TS mixture approach. *International Journal of Computer Integrated Manufacturing*, *14*(4), 367–384. https://doi.org/10.1080/0951120010020749

Chaouch, I., Driss, O. B., & Ghedira, K. (2017). A Survey of Optimization Techniques for Distributed Job Shop Scheduling Problems in Multi-factories. In R. Silhavy, R. Senkerik, Z. Kominkova Oplatkova, Z. Prokopova, & P. Silhavy (Eds.), *Cybernetics and Mathematics Applications in Intelligent Systems: Proceedings of the 6th Computer Science On-line Conference 2017 (CSOC2017), Vol 2* (pp. 369–378). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-57264-2_38

Chiang, T., & Fu, L. (2004). Solving the FMS scheduling problem by critical ratio-based heuristics and the genetic algorithm. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 3131–3136 Vol.3. https://doi.org/10.1109/ROBOT.2004.1307544

Daniels, R. L., & Chambers, R. J. (1990). Multiobjective flow-shop scheduling. In *Naval Research Logistics* (pp. 981–995).

Desprez, C., Chu, F., & Chu, C. (2009). Minimising the weighted number of tardy jobs in a hybrid flow shop with genetic algorithm. *International Journal of Computer Integrated Manufacturing*, *22*(8), 745–757. https://doi.org/10.1080/09511920902810938

Duan, W., Li, Z., Ji, M., Yang, Y., Wang, S., & Liu, B. (2016). A Hybrid Estimation of Distribution Algorithm for Distributed Permutation Flowshop Scheduling with Flowline Eligibility, (71101139), 2581–2587. https://doi.org/10.1109/CEC.2016.7744111

El Bouri, A., & Amin, G. R. (2015). A combined OWA-DEA method for dispatching rule selection. *Computers and Industrial Engineering*, *88*, 470–478. https://doi.org/10.1016/j.cie.2015.08.007

Fernandes, S., & Lourenço, H. (2007). A GRASP and branch-and-bound metaheuristic for the job-shop scheduling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4446 LNCS*, 60–71. https://doi.org/10.1007/978-3-540-71615-0_6

Framinan, J., & Leisten. (2003). An efficient constructive heuristic for flowtime minimisation in permutation flow shops. *Omega*, *31*(4), 311–317. https://doi.org/10.1016/S0305-0483(03)00047-1

Fuchigami, H. Y., & Rangel, S. (2017). A survey of case studies in production scheduling: Analysis and perspectives. *Journal of Computational Science*. https://doi.org/10.1016/j.jocs.2017.06.004

Gao, J., & Chen, R. (2011a). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem, (1), 3096–3101. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/18756891.2011.9727808#abstract

Gao, J., & Chen, R. (2011b). An NEH-based heuristic algorithm for distributed permutation flowshop scheduling problems. *Scientific Research and Essays*, *6*(14), 3094–3100. https://doi.org/10.5897/SRE10.1014

García, R. G. (2007). Integral optimization and some supply chain developments, 157.

Garza, R., González, C., Rodríguez, E., & Hernández, C. (2016). Application of Six Sigma DMAIC with Discrete Simulation and Multicriterial Techniques, (22), 19–36.

Geiger, C. D., & Uzsoy, R. (2006). RAPID MODELING AND DISCOVERY OF PRIORITY DISPATCHING RULES : AN AUTONOMOUS LEARNING, 7–34.

Gendreau, M., & Potvin, J.-Y. (2010). Handbook of Metaheuristics. In *Handbook of Metaheuristics* (2nd ed., pp. 283–319). Springer US. https://doi.org/10.1007/978-1-4419-1665-5

González, E., García, R. G., Caballero, J. P., Molina, L. P., & Montoya, J. (2016). Stochastic flexible flow shop scheduling problem under quantitative and qualitative decision criteria. *Computers & Industrial Engineering*, *101*, 128–144. https://doi.org/10.1016/j.cie.2016.08.026

González, E., & Montoya, J. (2017). A GRASP meta-heuristic for the hybrid flowshop scheduling problem. *Journal of Decision Systems*, *0125*(July), 1–13. https://doi.org/10.1080/12460125.2017.1351863

González, E., & Montoya, J. (2018). A simheuristic for stochastic permutation flow shop problem considering quantitative and qualitative decision criteria. *Proceedings of the 16th International Conference on Project Management and Scheduling*, 104–109.

González, E., Montoya, J., & Caballero, J. (2018). A comparison of dispatching rules hybridised with Monte Carlo Simulation in stochastic permutation flow shop problem. *Journal of Simulation*, *7778*(May), 1–10. https://doi.org/10.1080/17477778.2018.1473908

Goodman, A., & Hastak, M. (2015). Infrastructure Planning, Engineering, and Economics. In *Infrastructure Planning, Engineering, and Economics* (Second Edi). McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto. Retrieved from http://accessengineeringlibrary.com.ezproxy.javeriana.edu.co:2048/browse/infrastructure-planning-engineering-and-economics-second-edition#fullDetails

Haimes, Y. Y. (1998). Risk Modeling, Assessment, and Management. In *Risk Modeling, Assessment, and Management*.

Haupt, R. (1989). A survey of priority rule-based scheduling. *OR Spektrum*, *11*(1), 3–16. https://doi.org/10.1007/BF01721162

Hernández, M. del C. (2007). Introducción a la programación lineal. In *Introducción a la programación lineal* (UNAM, p. 9).

Ishibuchi, H., & Murata, T. (1998). A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *28*(3), 392–403. https://doi.org/10.1109/5326.704576

Jayamohan, M. S., & Rajendran, C. (2000). New dispatching rules for shop scheduling: A step forward. *International Journal of Production Research*, *38*(3), 563–586. https://doi.org/10.1080/002075400189301

Jung, H. (2010). An available-to-promise model considering customer priority and variance of penalty costs. *International Journal of Advanced Manufacturing Technology*, *49*(1–4), 369–377. https://doi.org/10.1007/s00170-009-2389-9

Knowles, J., & Corne, D. (1999). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, *1*(December 2013), 98–105. https://doi.org/10.1109/CEC.1999.781913

Lamothe, J., Marmier, F., Dupuy, M., Gaborit, P., & Dupont, L. (2012). Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints. *Computers and Operations Research*, *39*(6), 1236–1244. https://doi.org/10.1016/j.cor.2010.07.007

Lee, Y. H., Bhaskaran, K., & Pinedo, M. (1997). A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions (Institute of Industrial Engineers)*, *29*(1), 45–52. https://doi.org/10.1080/07408179708966311

Lin, D., Lee, C. K. M., & Wu, Z. (2012). Integrating analytical hierarchy process to genetic algorithm for re-entrant flow shop scheduling problem. *International Journal of Production Research*, *50*(7), 1813–1824. https://doi.org/10.1080/00207543.2011.561884

Liu, H., & Gao, L. (2010). A discrete electromagnetism-like mechanism algorithm for solving distributed permutation flowshop scheduling problem. *Proceedings - 2010 International Conference on Manufacturing Automation, ICMA 2010*, 156–163. https://doi.org/10.1109/ICMA.2010.17

Low, C., & Lin, W. Y. (2011). Minimizing the total completion time in a single-machine scheduling problem with a learning effect. *Applied Mathematical Modelling*, *35*(4), 1946–1951. https://doi.org/10.1016/j.apm.2010.11.006

Lu, P.-H., Tan, H., Peng, Y.-H., Chen, C.-F., & Wu, M.-C. (2014). A new solution representation for developing meta-heuristic algorithms to solve distributed flexible job-shop scheduling problems, 1009–

1015.
McGovern, S., & Gupta, S. (2011). Combinatorial Optimization Searches. In *Disassembly Line: Balancing and Modeling* (The McGraw, p. Chapter 10. Combinatorial Optimization Searches). New York, Chicago, San Francisco, Lisbon, London, Madrid, Mexico City, Milan, New Delhi, San Juan Seoul, Singapore, Sydney, Toronto. Retrieved from http://accessengineeringlibrary.com.ezproxy.javeriana.edu.co:2048/browse/disassembly-line-balancing-and-modeling/p2001c2f19970117001?q=+Iterative+Computer+Algorithms+with+Applications+in+Engineering%3A+Solving+Combinatorial+Optimization+Problems
Naderi, B., & Ruiz, R. (2010). The distributed permutation flowshop scheduling problem. *Computers & Operations Research*, *37*(4), 754–768. https://doi.org/10.1016/j.cor.2009.06.019
Naderi, B., & Ruiz, R. (2014). A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, *239*(2), 323–334. https://doi.org/10.1016/j.ejor.2014.05.024
Niebel, B., & Freivalds, A. (2009). Ingeniería Industrial: Métodos, estándares y diseño de trabajo. 12 ed. In *Ingeniería Industrial: Métodos, estándares y diseño de trabajo. 12 ed* (The McGraw, p. 348).
Pinilla, A., & Castro, A. (2015). Optimización multiobjetivo en la gestión de cadenas de suministro de biocombustibles . Una revisión de la literatura Multiobjective optimization in biofuel supply chain management . A review of the literature, 2–3.
Ptak, C., & Chad, S. (2011). Master Production Schedule. In *Orlicky's Material Requirements Planning, Third Edition* (McGraw-Hil, p. 12. Master Production Schedule). McGraw-Hill Education: New York, Chicago, San Francisco, Lisbon, London, Madrid, Mexico City, Milan, New Delhi, San Juan, Seoul, Singapore, Sydney, Toronto.
Rajendran, C., & Holthaus, O. (1999). Comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operational Research*, *116*(1), 156–170. https://doi.org/10.1016/S0377-2217(98)00023-X
Resende, M. (1998). Greedy randomized adaptive search procedures (grasp), 1–11.
Rodríguez, C. F. (2017). *Entrevista personal*.
Rojanasoonthon, S., & Bard, J. (2005). A GRASP for Parallel Machine Scheduling with Time Windows. *INFORMS Journal on Computing*, *17*(1), 32–51. https://doi.org/10.1287/ijoc.1030.0048
Roychowdhury, S., Allen, T. T., & Allen, N. B. (2017). A genetic algorithm with an earliest due date encoding for scheduling automotive stamping operations. *Computers and Industrial Engineering*, *105*, 201–209. https://doi.org/10.1016/j.cie.2017.01.007
Saaty, T. L. (1990). How to make decision: The Analytic Hierarchy Process. *Eoropean Journal of Operation Research*. https://doi.org/10.1016/0377-2217(90)90057-I
Sait, S., & Youssef, H. (2000). Combinatorial Optimization. In *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems* (Wiley-IEEE).
Savelsbergh, M., Uma, R., & Wein, J. (2005). An Experimental Study of LP-Based Approximation Algorithms for Scheduling Problems. *INFORMS Journal on Computing*, *17*(1), 123–136. https://doi.org/10.1287/ijoc.1030.0055
Shafaei, R., & Brunn, P. (1999). Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, *37*(17), 3913–3925. https://doi.org/10.1080/002075499189835
Zandin, K. (2001). Maynard's Industrial Engineering Handbook. In *Maynard's Industrial Engineering Handbook* (Fifth Edit). McGRAW-HILL: New York, Chicago, San Francisco, Lisbon, London Madrid, Mexico City, Milan, New Delhi, San Juan Seoul, Singapore, Sydney, Toronto. Retrieved from http://accessengineeringlibrary.com.ezproxy.javeriana.edu.co:2048/browse/maynards-industrial-engineering-handbook-fifth-edition/p2000a1fc9979.143001